Wykład nr 1
Temat: Wprowadzenie do programowania. Algorytm, program, kompilator, interpreter. Wybór i konfiguracja środowiska programistycznego

Paweł Karczmarek
Katedra Inteligencji Obliczeniowej Politechniki Lubelskiej

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Plan prezentacji

- Algorytm
- Program
- Kompilator
- Interpreter
- Środowisko programistyczne

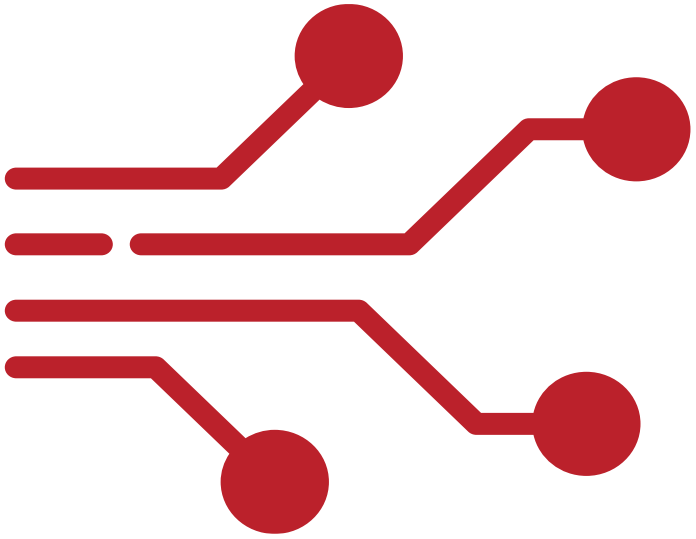
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

2

Czym jest programowanie?

Bardzo ogólna definicja



Programowanie to proces tworzenia programów komputerowych – czyli zestawu instrukcji, które komputer może wykonać w celu rozwiązania określonego problemu.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

3

Dlaczego warto uczyć się programowania?

- Jest to jedna z najbardziej poszukiwanych umiejętności na rynku pracy.
- Programiści są potrzebni w wielu branżach (IT, finanse, przemysł, medycyna, edukacja).
- Pozwala automatyzować żmudne i powtarzalne zadania.
- Daje dostęp do dobrze płatnych stanowisk.
- Rozwija logiczne i analityczne myślenie.
- Uczy systematycznego rozwiązywania problemów.
- Pozwala na twórczość i kreatywność (gry, aplikacje, systemy).
- Daje satysfakcję z tworzenia działających rozwiązań.
- Jest uniwersalne – przydatne w naukach ścisłych, biznesie i sztuce.
- Pomaga lepiej rozumieć współczesny świat technologii.

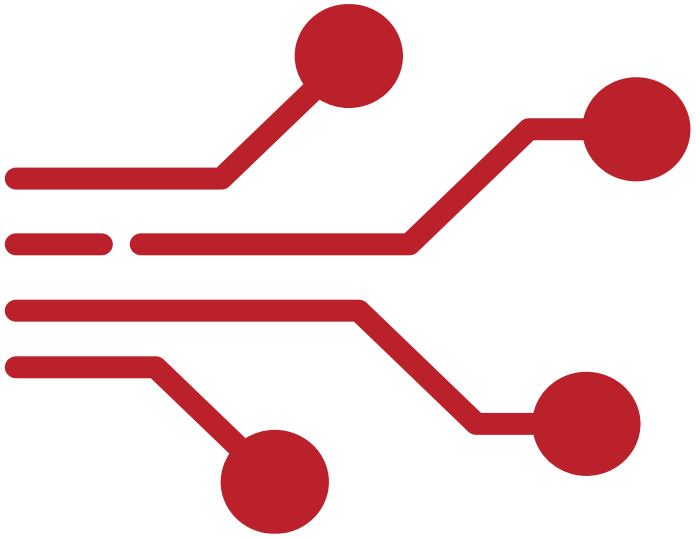
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

4

Programowanie a AI

- Czy AI zastąpi programistów?
- Co z rynkiem pracy juniorów?
- Czy wciąż warto programować?



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

5

Zastosowania programowania w życiu codziennym i pracy

- Życie codzienne:
  - Aplikacje mobilne: komunikatory, bankowość, mapy, zakupy online.
  - Gry komputerowe i multimedia.
  - Urządzenia IoT w domu (inteligentne żarówki, robot sprzątający).
  - Aplikacje do zdrowia i sportu (smartwatch, fitness).
- Praca zawodowa
  - Systemy biznesowe (ERP, CRM, bankowe).
  - Automatyzacja procesów (raporty, faktury, workflow).
  - Analiza danych i sztuczna inteligencja.
  - Sterowanie urządzeniami i robotyką w przemyśle.
  - Medycyna – oprogramowanie diagnostyczne i sprzęt medyczny.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

6

Algorytm

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

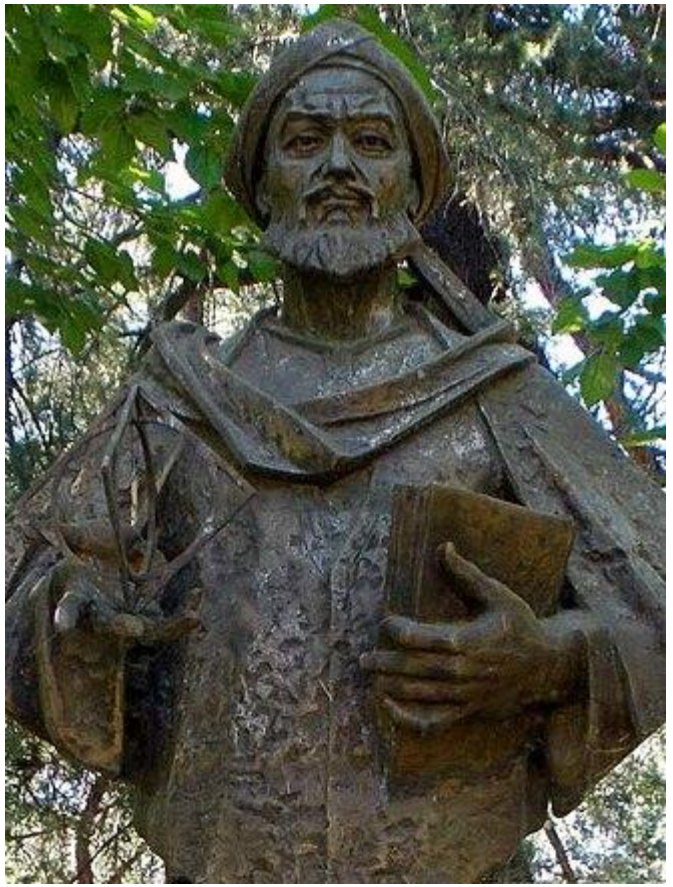
Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

7

Algorytm (definicja wg Donalda Knutha)

Algorytm jest skończonym, jednoznacznie określonym zestawem instrukcji, które w skończonym czasie prowadzą do rozwiązania problemu

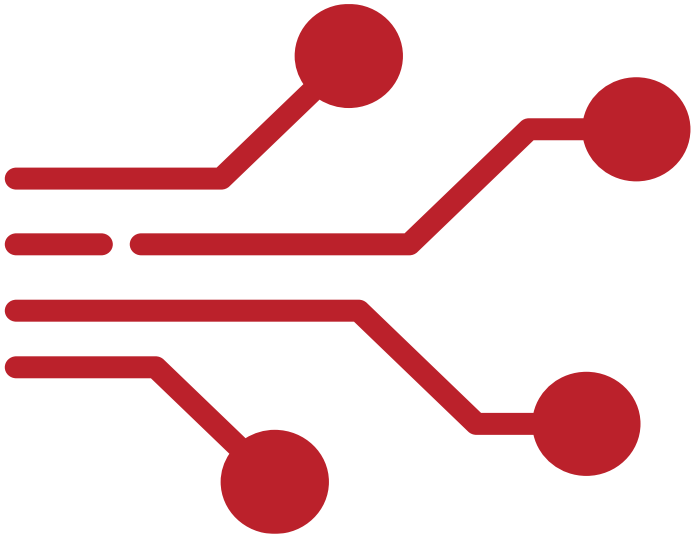
Rys. 1. Donald Knuth [źródło: Wikipedia]



Rys. 2. Muhammad ibn Musa al-Chuwarizmi [źródło: Wikipedia]

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

8



Cechy algorytmu:

- Skończoność
- Jednoznaczność
- Efektywność
- Uniwersalność
- Poprawność

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

9

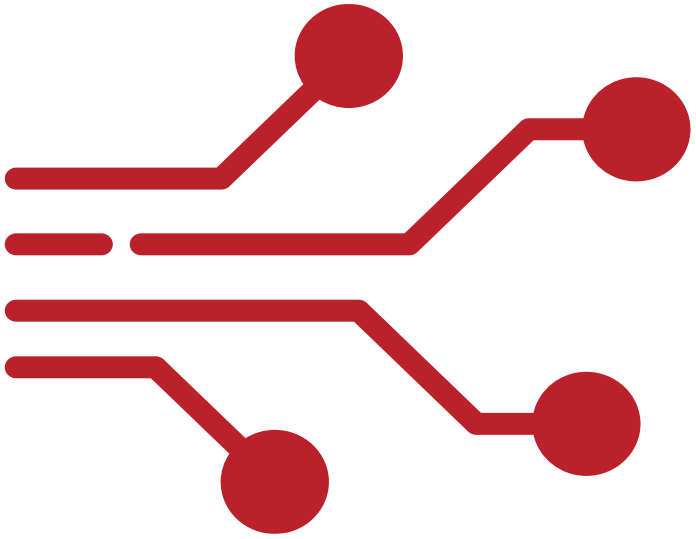
Algorytmy w życiu codziennym

- Przepisy kulinarne
- Instrukcje obsługi
- Plany działania
- Codzienne czynności

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

10



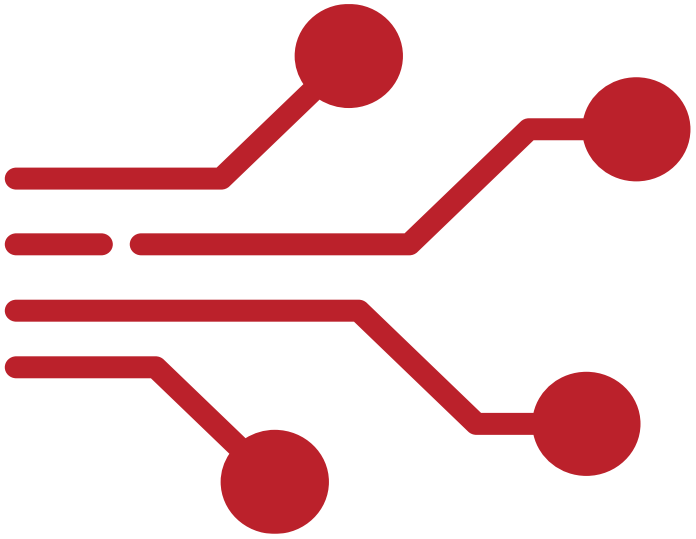
Algorytmy w informatyce

- Sortowania
- Wyszukiwania
- Grafowe
- Kryptograficzne
- Sztucznej inteligencji

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

11



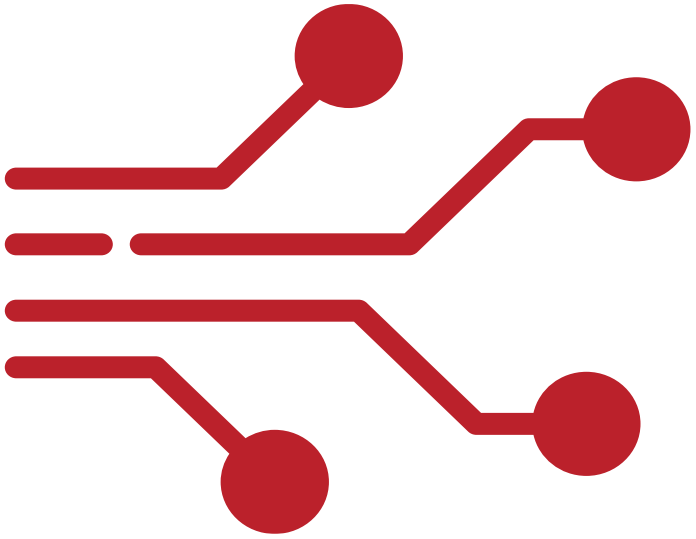
Klasy algorytmów używanych w cyberbezpieczeństwie

- Szyfrowanie symetryczne
- Szyfrowanie asymetryczne
- Funkcje haszujące
- Podpisy cyfrowe
- Protokoły bezpieczeństwa

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

12



Sposoby przedstawienia algorytmu:

- Tekst pisany (opis słowny)
- Tekst ustrukturyzowany
- Pseudokod
- Zapis graficzny
  - Schemat blokowy
  - Schemat N-S (Nassi-Shneidermana)
  - Diagram czynności UML
- Kod źródłowy

...

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

13

Przykładowy pseudokod

```
ALGORYTM SORTOWANIE_KOLEJNYCH_MINIMÓW(A)
    n ← długość(A)

    DLA i od 0 do n-2 WYKONAJ
        min ← i

        DLA j od i+1 do n-1 WYKONAJ
            JEŻELI A[j] < A[min] TO
                min ← j
            KONIEC JEŻELI
        KONIEC DLA

        JEŻELI min ≠ i TO
            zamień A[i] z A[min]
        KONIEC JEŻELI
    KONIEC DLA

    ZWRÓĆ A
KONIEC ALGORYTMU
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

14

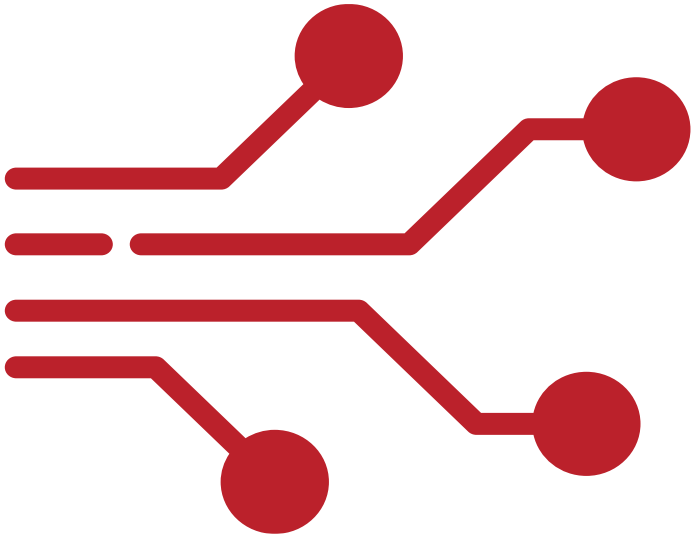
Przykładowy schemat blokowy

Rys. 3. Przykładowy schemat blokowy [źródło: Wikipedia]

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

15



Elementy

a) blok graniczny – początek, koniec, przerwanie lub wstrzymanie wykonywania działania

b) blok wejścia-wyjścia

c) blok operacyjny

d) blok decyzyjny, warunkowy

e) blok wywołania podprogramu

f) blok fragmentu

g) blok komentarza

h) łącznik wewnętrzny

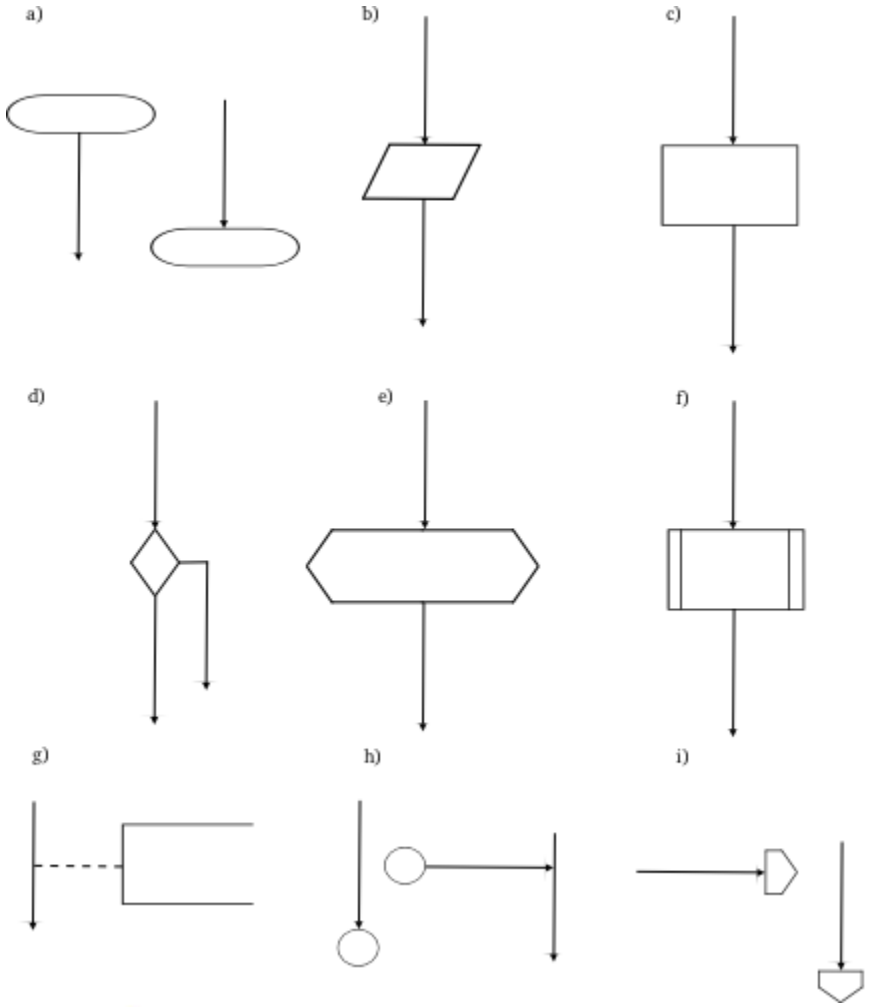
i) łącznik zewnętrzny

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

16

Elementy



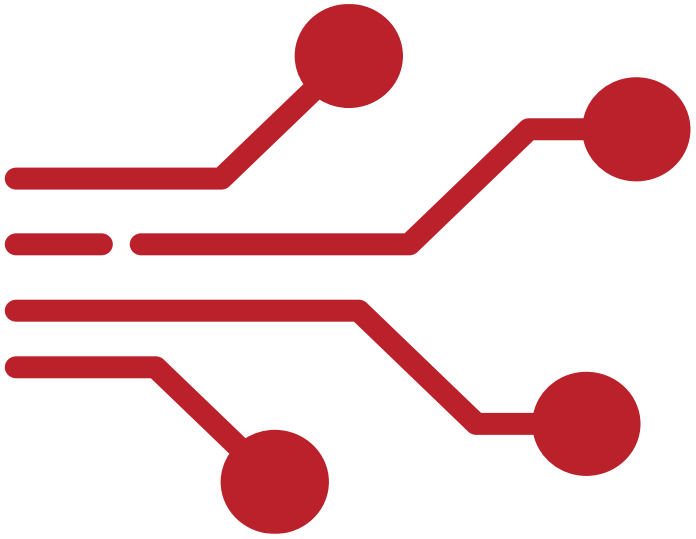
Rys. 4. Składniki schematu blokowego [źródło: Wikipedia]

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

17

Zamiana liczb A i B – tekst ustrukturyzowany

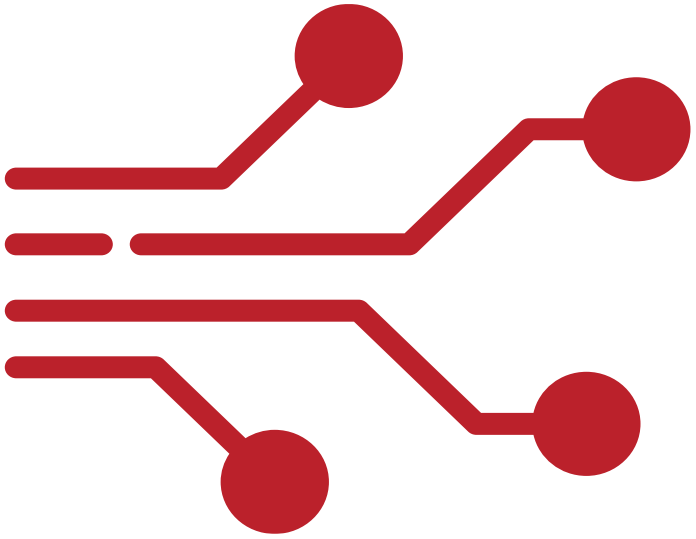


1. Wczytaj dwie liczby A i B.
2. Utwórz zmienną pomocniczą T.
3. Przypisz T <- A.
4. Przypisz A <- B.
5. Przypisz B <- T.
6. Zakończ (wartości A i B zostały zamienione).

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

18

Zamiana liczb A i B – pseudokod



WEJŚCIE: A, B

WYJŚCIE: A, B (po zamianie wartości)

T <- A

A <- B

B <- T

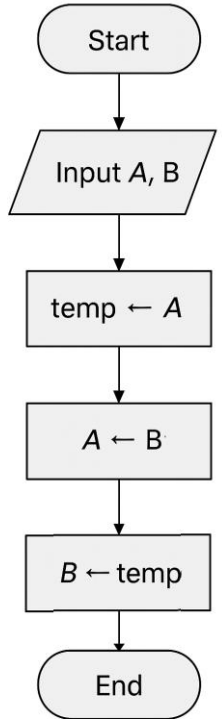
ZWRÓĆ A, B

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

19

Zamiana liczb A i B – schemat blokowy

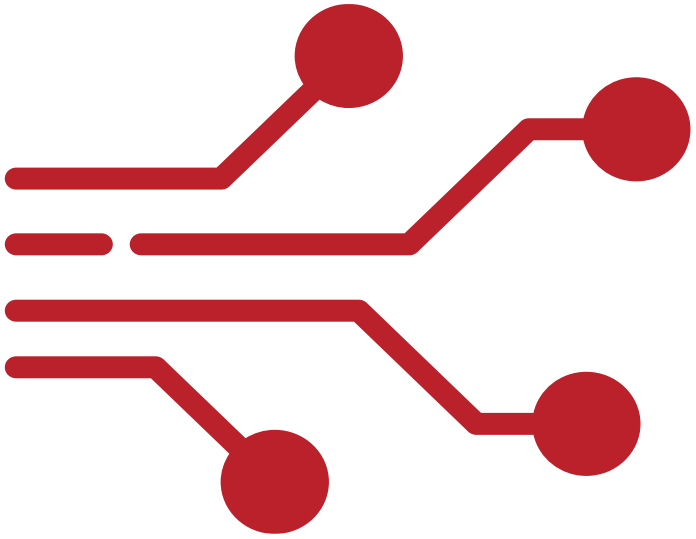


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

20

Zamiana liczb A i B – kod źródłowy



```
int A = 3;
int B = 7;

int T = A;
A = B;
B = T;
// A=7, B=3
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

21

Do przemyślenia:

Jak zamienić dwie liczby nie korzystając z funkcji swap lub zmiennej pomocniczej?

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

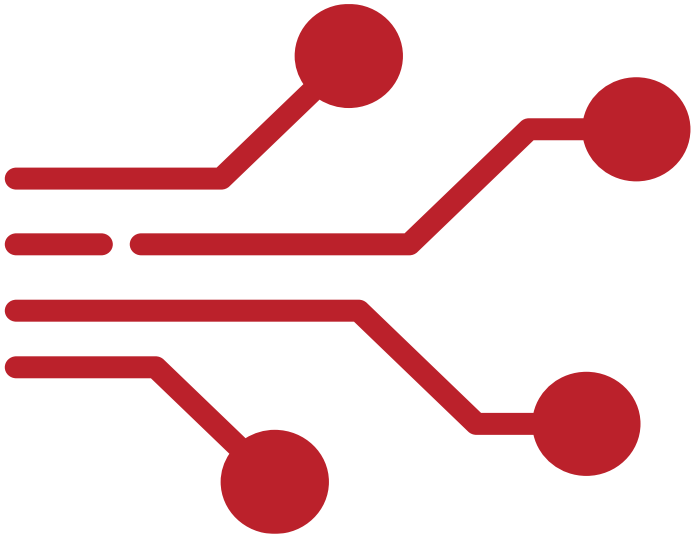
22

Program

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

23



Program

Zbiór instrukcji zapisanych w określonym języku programowania. Instrukcje te są wykonywane przez komputer w celu rozwiązania problemu lub realizacji zadania.

Charakterystyka programu

- Ma dane wejściowe (input).
- Przetwarza je zgodnie z algorytmem.
- Generuje wynik (output).

Elementy programu

- Instrukcje (polecenia dla komputera).
- Struktury danych.
- Komentarze i dokumentacja.

24

Inne ujęcie: Program komputerowy = algorytm + zapis w języku programowania.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Poziomy języków programowania

- Języki niskiego poziomu
  - Bliskie maszynie, trudniejsze do zrozumienia dla człowieka.
  - Operują bezpośrednio na rejestrach i pamięci.
  - Przykłady: język maszynowy, asembler.
  - Zalety: wysoka wydajność, pełna kontrola nad sprzętem.
  - Wady: trudne do pisania i utrzymania, mała przenośność.
- Języki wysokiego poziomu
  - Bliskie człowiekowi, oddalone od architektury sprzętu.
  - Zawierają struktury kontrolne, typy danych, biblioteki.
  - Przykłady: C, Java, Python, C#, JavaScript.
  - Zalety: czytelność, szybkie tworzenie aplikacji, przenośność.
  - Wady: mniejsza kontrola nad sprzętem, wolniejsze od niskopoziomowych.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

25

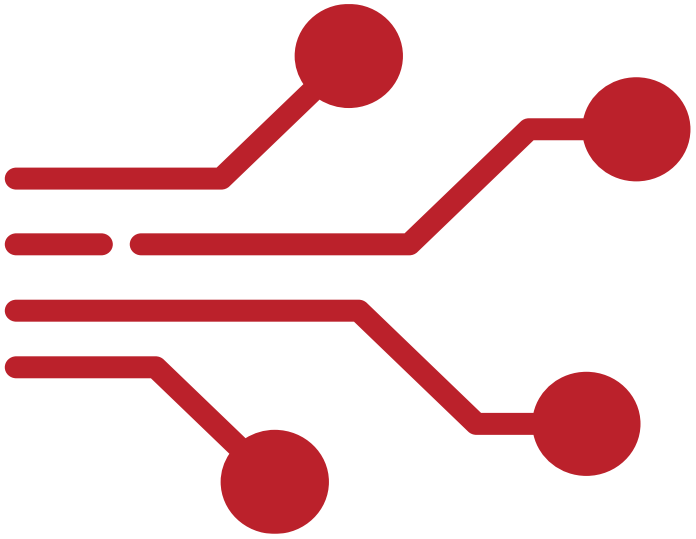
Przykłady

- Języki niskiego poziomu
  - Asembler – programowanie blisko sprzętu, systemy wbudowane.
- Języki ogólnego przeznaczenia
  - C – systemy operacyjne, sterowniki, oprogramowanie niskopoziomowe.
  - C++ – aplikacje wydajne, gry, systemy czasu rzeczywistego.
  - Java – aplikacje biznesowe, systemy korporacyjne, Android.
  - C# – aplikacje desktopowe i webowe, gry w Unity.
  - Python – AI, analiza danych, aplikacje webowe, skrypty automatyzujące.
  - JavaScript – aplikacje webowe (frontend i backend).
- Języki specjalizowane
  - R – analiza danych, statystyka.
  - MATLAB – obliczenia inżynierskie i naukowe.
  - SQL – bazy danych.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

26



Struktura prostego programu

- Elementy wejściowe
  - Deklaracje zmiennych.
  - Instrukcje wczytywania danych od użytkownika.
- Część przetwarzająca
  - Instrukcje wykonujące obliczenia.
  - Instrukcje warunkowe i pętle.
- Elementy wyjściowe
  - Instrukcje wyświetlania wyników.
  - Zapis danych do pliku lub bazy danych.
- Przykład minimalny („Hello, World!”)
  - Program bez danych wejściowych.
  - Instrukcja wyjścia: wyświetlenie tekstu na ekranie.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

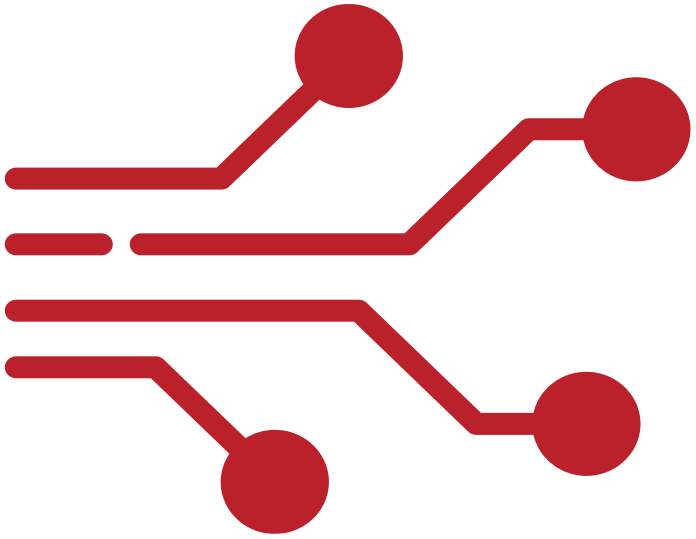
27

Cykl życia programu

- Tworzenie: Analiza problemu i zaprojektowanie algorytmu. Napisanie kodu źródłowego w wybranym języku.
- Kompilacja / interpretacja: Przetłumaczenie kodu źródłowego na język maszynowy lub pośredni. Wykrywanie błędów składniowych.
- Uruchamianie: Wykonanie programu na komputerze. Testowanie działania na danych przykładowych.
- Poprawki i rozwój: Usuwanie błędów (debugowanie). Dodawanie nowych funkcji.
- Utrzymanie: Dbanie o zgodność programu z nowym sprzętem, systemem i wymaganiami użytkowników.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

28



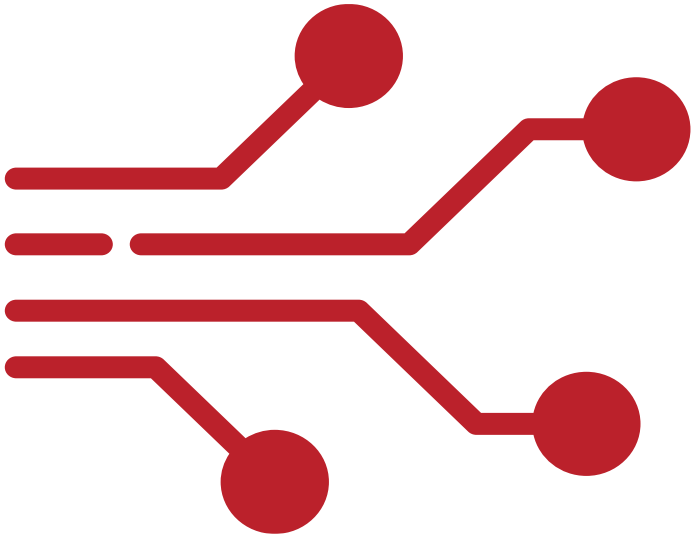
Kompilator

- Kompilator to program służący do automatycznego tłumaczenia kodu napisanego w jednym języku (języku źródłowym) na równoważny kod w innym języku (języku wynikowym).
- Proces ten nazywany jest kompilacją.
- W informatyce kompilatorem nazywa się najczęściej program do tłumaczenia kodu źródłowego w języku programowania na język maszynowy. Niektóre z kompilatorów tłumaczą najpierw do języka asemblera, a ten na język maszynowy jest tłumaczony przez asembler.
- Etapy kompilacji (w uproszczeniu)
  - Analiza leksykalna (rozpoznanie słów kluczowych i symboli).
  - Analiza składniowa (sprawdzenie poprawności struktury kodu).
  - Generowanie kodu wynikowego (język maszynowy/bytecode).

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

29



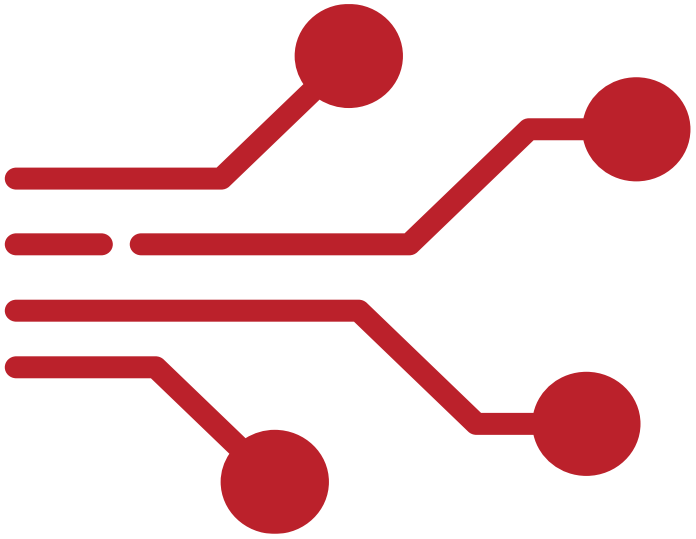
Zalety i wady kompilacji

- Zalety
  - Program po kompilacji działa szybciej (kod maszynowy wykonywany bezpośrednio).
  - Kod źródłowy nie musi być rozpowszechniany – użytkownik otrzymuje plik wykonywalny.
  - Możliwość optymalizacji kodu przez kompilator.
  - Lepsza kontrola nad błędami składniowymi (wykrywane przed uruchomieniem).
- Wady
  - Proces kompilacji zajmuje dodatkowy czas.
  - Każda zmiana w kodzie wymaga ponownej kompilacji.
  - Plik wykonywalny jest zależny od platformy (systemu i architektury procesora).

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00
Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Interpreter

to program, który wykonuje kod źródłowy bez wcześniejszej kompilacji do pliku maszynowego. Tłumaczy i uruchamia instrukcje programu linia po linii.



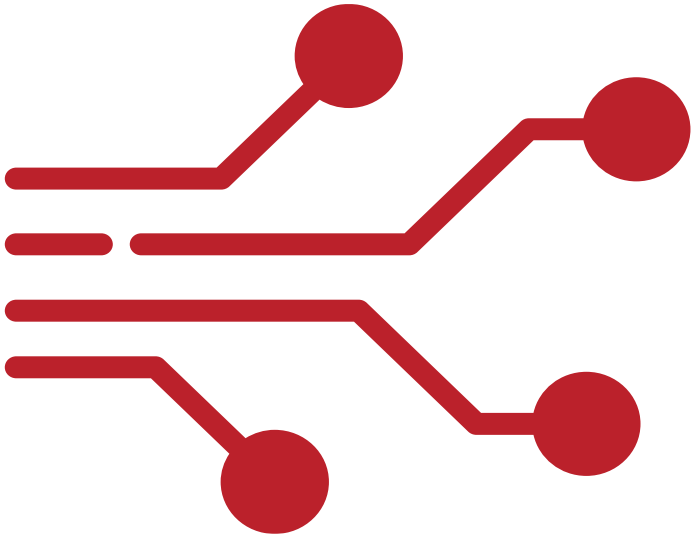
Charakterystyka

- Nie tworzy pliku wykonywalnego.
- Program działa od razu po uruchomieniu w interpreterze.
- Wykrywa błędy dopiero w momencie wykonywania danej instrukcji.

Przykłady interpreterów

- CPython (dla języka Python).
- Node.js (dla JavaScript).
- PHP interpreter.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00



Zalety i wady interpretacji

- Zalety
  - Program działa od razu, bez potrzeby kompilacji.
  - Łatwo testować i poprawiać kod fragmentami (np. w trybie REPL).
  - Wygodne w nauce i szybkim prototypowaniu.
  - Przenośność — ten sam kod można uruchomić na różnych systemach, jeśli dostępny jest interpreter.
- Wady
  - Wolniejsze działanie w porównaniu do programów skompilowanych.
  - Błędy wychodzą dopiero podczas uruchamiania programu.
  - Konieczność posiadania interpretera na każdym urządzeniu.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Języki kompilowane/interpretowane

Języki kompilowane

- Kod źródłowy tłumaczony na kod maszynowy przed uruchomieniem.
- Program działa szybciej.
- Błędy składniowe wykrywane na etapie kompilacji.
- Przykłady: C, C++, Rust, Go.

Języki interpretowane

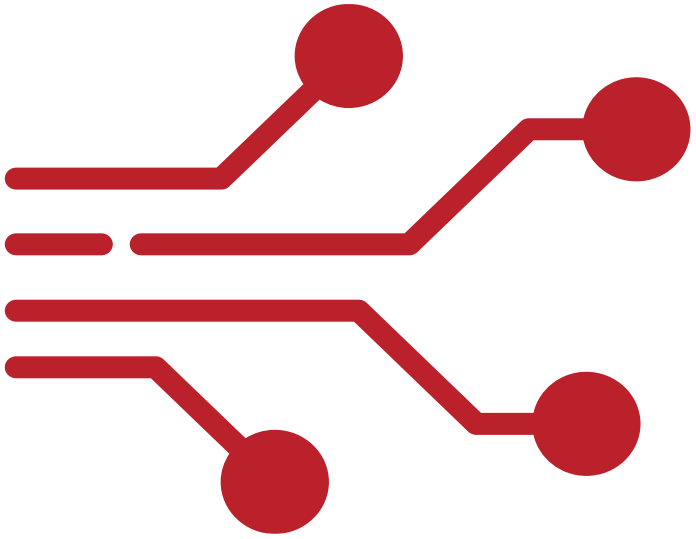
- Kod źródłowy tłumaczony i wykonywany „w locie”.
- Łatwiejsze testowanie i prototypowanie.
- Wolniejsze działanie programu.
- Przykłady: Python, JavaScript, PHP, Ruby.

W praktyce wiele języków korzysta z podejścia mieszanego (kompilacja + interpretacja).

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

33



Języki hybrydowe (kompilacja + interpretacja, JIT)

- Idea hybrydy: Kod źródłowy nie jest tłumaczony bezpośrednio na język maszynowy. Najpierw kompilacja do kodu pośredniego (bytecode). Następnie uruchamiany przez wirtualną maszynę (interpreter + JIT).
- JIT (Just-In-Time Compilation)
  - Fragmenty bytecode tłumaczone dynamicznie na kod maszynowy.
  - Łączy zalety interpretacji (elastyczność) i kompilacji (szybkość).

Przykłady

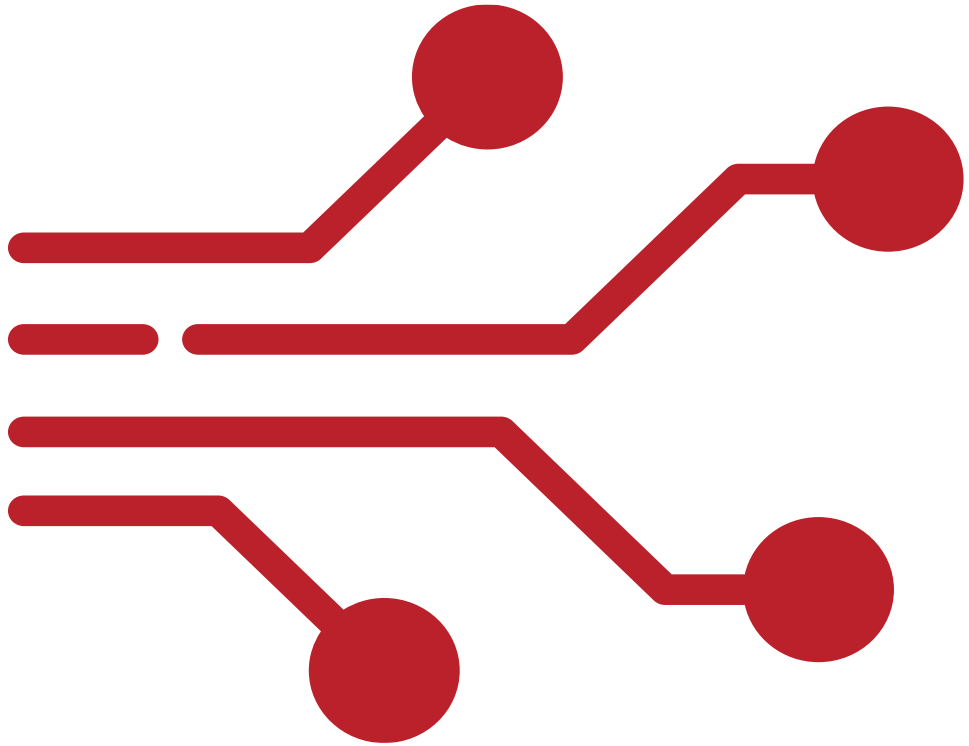
- Java -> bytecode wykonywany na JVM.
- C# -> kod IL (Intermediate Language) uruchamiany w CLR/.NET.
- Python (PyPy) -> interpreter z JIT.

Zalety

- Dobra przenośność (ten sam bytecode działa na różnych systemach).
- Możliwość optymalizacji w trakcie działania programu.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00
Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Środowisko programistyczne



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

35

Zintegrowane środowisko programistyczne (IDE – Integrated Development Environment)

- Zbiór narzędzi ułatwiających tworzenie, testowanie i uruchamianie programów.
- Łączy w jednym miejscu edytor kodu, kompilator/interpreter i narzędzia pomocnicze.

Podstawowe elementy IDE

- Edytor kodu (podświetlanie składni, autouzupełnianie).
- Kompilator lub interpreter.
- Debugger (śledzenie działania programu).
- Konsola lub terminal.

Zadania środowiska programistycznego

- Ułatwia pisanie kodu i wykrywanie błędów.
- Przyspiesza proces tworzenia oprogramowania.
- Zapewnia integrację z systemami kontroli wersji i menedżerami pakietów.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

36

Edytory tekstowe/IDE

Proste programy do pisania kodu.

Lekkie, szybkie, często rozszerzalne wtyczkami.

Przykłady: Notepad++, Sublime Text, Vim, nano.

Zalety: małe zużycie zasobów, szybkość działania.

Wady: brak wbudowanego kompilatora/debuggera (trzeba uruchamiać osobno).

Rozbudowane środowiska integrujące wiele narzędzi.

Zawierają edytor kodu, kompilator/interpreter, debugger, menedżery projektów.

Przykłady: Visual Studio, IntelliJ IDEA, Eclipse, PyCharm.

Zalety: wygoda, automatyzacja, podpowiedzi składni, integracja z Git.

Wady: większe wymagania sprzętowe, dłuższy czas uruchamiania.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Konfiguracja środowiska Visual Studio

- Pobranie i instalacja: [visualstudio.microsoft.com]. Edycja darmowa to Community. Podczas instalacji wybieramy Workload -> .NET desktop development.
- Tworzenie nowego projektu: Visual Studio -> File -> New -> Project. Szablon: Console App (.NET). Nadajemy nazwę projektowi i wybieramy folder.
- Struktura projektu: Plik główny Program.cs zawiera funkcję Main(). Folder Dependencies – biblioteki systemowe i zewnętrzne. Plik konfiguracyjny .csproj.
- Uruchamianie programu: zielony przycisk Run lub Ctrl + F5. Wynik pojawi się w konsoli.
- Debugowanie: Ustawiamy breakpoint (czerwona kropka obok linii kodu). F5, aby uruchomić program w trybie debugowania. Podglądamy wartości zmiennych w oknie Locals.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

38

Kod programu

- Klasycznie (z Main):

```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Hello, World!");
    }
}
```

- Nowocześnie: z top-level statements (C# 9+)

```
Console.WriteLine("Hello, World!");
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

39

.NET

- Platforma programistyczna opracowana przez Microsoft. Umożliwia tworzenie i uruchamianie aplikacji w różnych językach (np. C#, F#, VB.NET).

Główne elementy:

- CLR (Common Language Runtime) – środowisko uruchomieniowe:
  - zarządza pamięcią,
  - obsługuje błędy i bezpieczeństwo,
  - wykonuje kod w postaci IL (Intermediate Language).
- Biblioteki klas .NET – gotowe moduły do pracy z plikami, bazami danych, siecią, grafiką.

Rodzaje .NET

- .NET Framework – starsza wersja, głównie Windows.
- .NET (Core) – nowoczesna, otwarta, wieloplatformowa (Windows, Linux, macOS).

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

40

IL (Intermediate Language, język pośredni)

Kod generowany podczas kompilacji programów .NET (C#, F#, VB.NET).

- Niezależny od systemu i sprzętu
- Rola w uruchamianiu programu

1. Kod źródłowy (np. w C#) -> kompilacja do IL.
2. Plik .exe lub .dll zawiera IL + metadane.
3. CLR (Common Language Runtime) tłumaczy IL na kod maszynowy procesora (JIT).

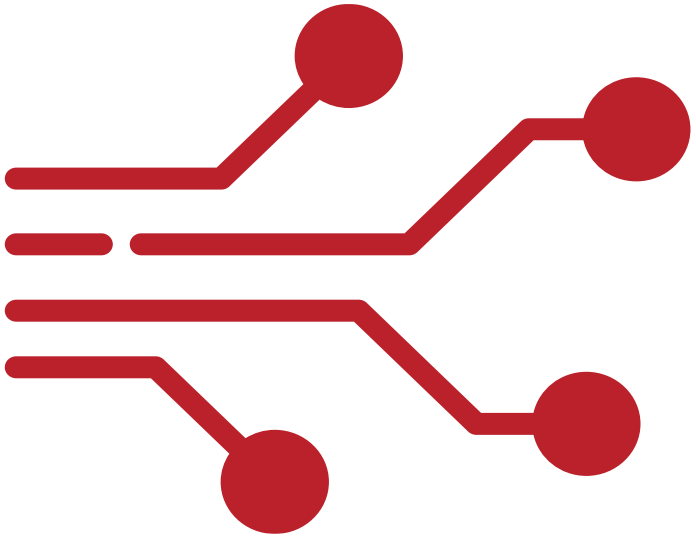
Zalety

- Przenośność – Windows, Linux, macOS.
- Współpraca wielu języków .NET (np. C# + F# + VB.NET).
- Optymalizacja w trakcie działania programu.

Przykład IL dla Console.WriteLine("Hello, World!");

```
IL_0001: ldstr "Hello, World!"
IL_0006: call void [System.Console]System.Console::WriteLine(string)
IL_000B: ret
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05/IP.08-0[illegible]/20.



2000: Microsoft ogłasza C# 1.0 wraz z .NET Framework 1.0. Cel: konkurencja dla Javy. Podstawowe elementy: programowanie obiektowe, typy wartościowe i referencyjne.

2005: C# 2.0. Wprowadzenie typów ogólnych (generics). Udoskonalenia w delegatach i iteracji.

2007: C# 3.0. Dodanie LINQ (Language Integrated Query). Wyrażenia lambda, metody rozszerzające.

2010: C# 4.0. Wsparcie dla programowania dynamicznego (dynamic). Ułatwienia w interoperacyjności z COM i Office.

2012: C# 5.0. Dodanie async/await – asynchroniczne programowanie w prosty sposób.

2015: C# 6.0. Nowa składnia (interpolacja stringów, wyrażenia w ciele właściwości).

2017: C# 7.x. Krotki, lokalne funkcje, dopasowanie do wzorca (pattern matching).

2020: C# 9.0. Top-level statements (pisanie kodu bez Main). Rekordy (records).

2021–2023: C# 10/11/12. Uproszczenia składni, globalne usingi, rekordy strukturalne. Rozwój w kierunku większej czytelności i nowoczesności.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Dziękuję za uwagę

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Wykład nr 2
Temat: Podstawy składni. Zmienne i typy danych. Operatory arytmetyczne i logiczne

Paweł Karczmarek
Katedra Inteligencji Obliczeniowej Politechniki Lubelskiej

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Plan prezentacji

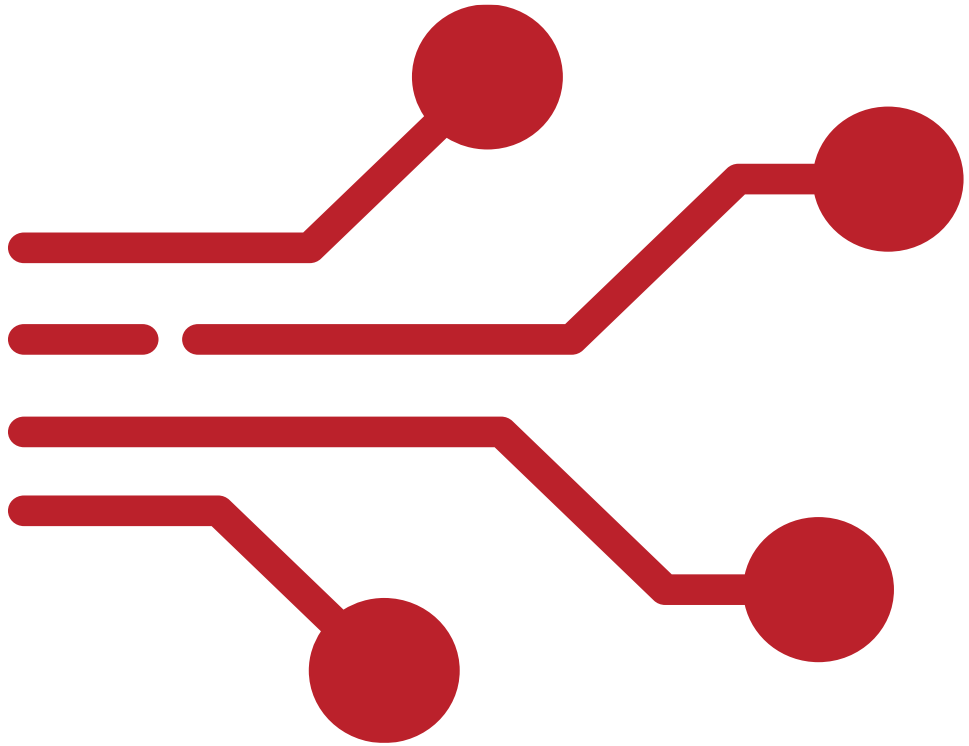
- Podstawy składni
- Zmienne i typy danych
- Typy i proste literały
- Operatory arytmetyczne
- Operatory logiczne i warunkowe

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

45

Podstawy składni

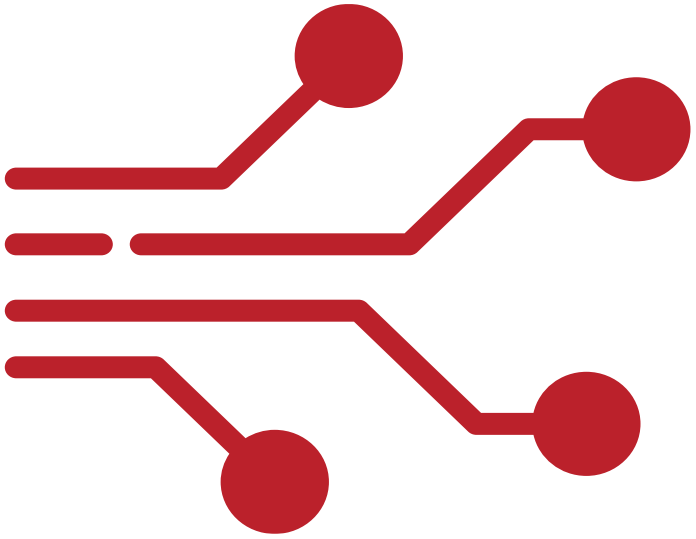


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

46

Hello World!

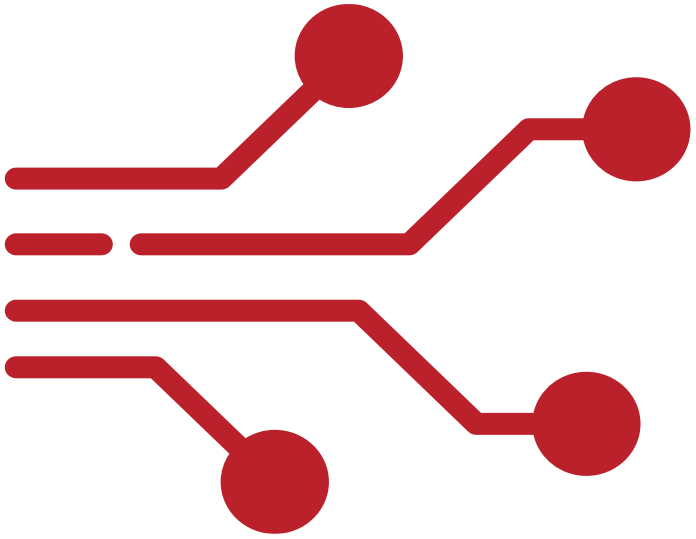
```
using System;
namespace Hello
{
        class Program
        {
                static void Main(string[] args)
                {
                        Console.WriteLine("Hello World!");
                }
        }
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

47



Przydatne skróty klawiszowe w VS

Ctrl+F Przeszukiwanie kodu

Ctrl+L Usunięcie bieżącej linii

F5 Kompilacja i uruchomienie aplikacji w trybie debugowania

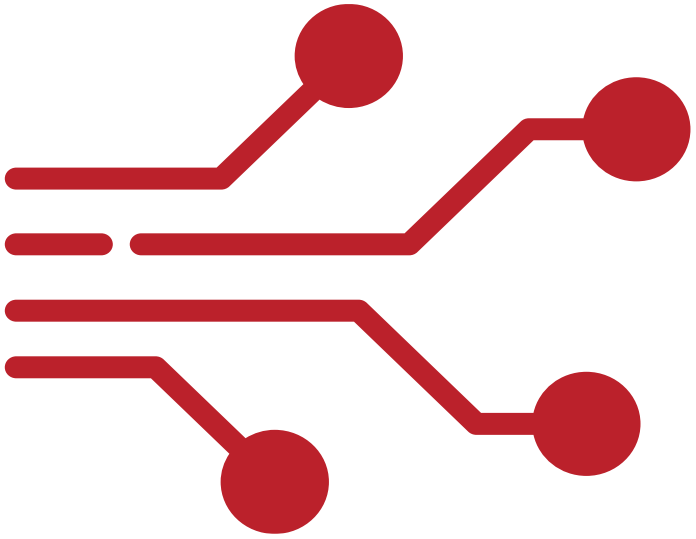
Ctrl+F5 Kompilacja i uruchomienie aplikacji bez debugowania

Ctrl+K+C/U – komentowanie bloku/odkomentowanie bloku

Ctrl+M+O – zwijanie kodu

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

48



Podstawowa składnia

; kończy instrukcję

Identyfikatory – nazwy nadane przez programistę

Słowa kluczowe – nazwy o specjalnym znaczeniu dla kompilatora

Literały – podstawowe porcje danych wprowadzone do kodu źródłowego

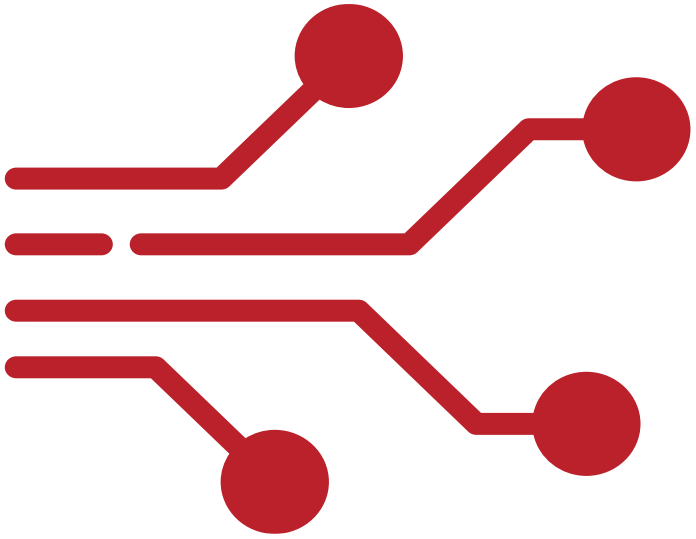
Znaki interpunkcyjne – określają strukturę programu

Operatory – przekształcają wyrażenia lub tworzą ich kombinacje

Komentarze: // - jednowierszowe, /* … */ - wielowierszowe

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

49



Bloki kodu

- Przestrzeń nazwy – zawiera i grupuje typy
- Klasa – zawiera elementy obiektu, w tym metody
- Metoda – zawiera instrukcje implementujące działania wykonywane przez dany obiekt
- Region – do definiowania własnych bloków kodu

```
#region mojregion
fragment kodu
#endregion
```

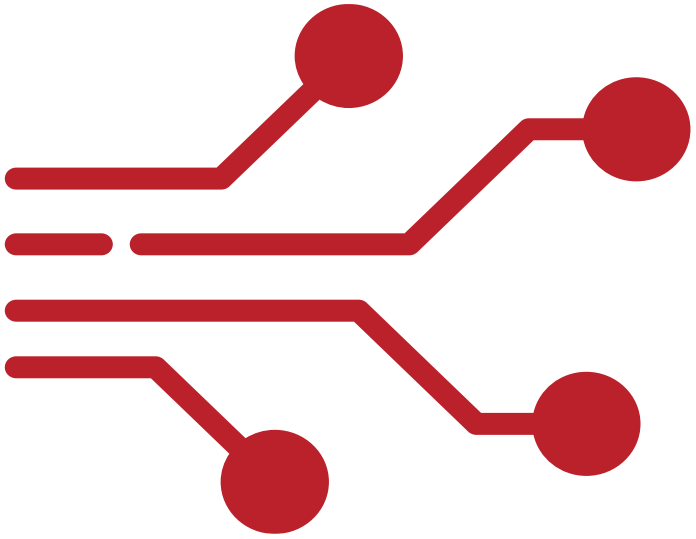

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

50

Importowanie przestrzeni nazw

Na przykładzie:



Można powiedzieć, że System jest przestrzenią nazw, czyli czymś w rodzaju adresu dla typów.

Nazwa System.Console.WriteLine informuje kompilator, że ma poszukiwać metody WriteLine z typu o nazwie Console w przestrzeni nazw System.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

51

Konwencje nazewnictwa

- Zasady ogólne
  - Czytelność ponad wszystko – nazwy muszą jasno oddawać przeznaczenie.
  - Język angielski – standard w kodzie (ułatwia współpracę międzynarodową).
  - Unikanie skrótów (poza powszechnie znanymi: id, URL, XML).
- Styl pisowni
  - PascalCase - klasy, struktury, interfejsy, przestrzenie nazw, metody, właściwości, stałe (Person, GetName()).
  - camelCase - zmienne lokalne, parametry metod (firstName, counter).
  - ALL_CAPS - stałe symboliczne, zwykle w kodzie niskopoziomowym (MAX_SIZE).
  - I - prefiks dla interfejsów (IEnumerable, IDisposable).
  - _ - podkreślnik dla zmiennych globalnych w klasach
- Dobre praktyki
  - Nazwy rzeczownikowe dla typów (Customer, Invoice).
  - Nazwy czasownikowe dla metod (CalculateSum(), PrintReport()).
  - Unikanie polskich znaków i znaków specjalnych.
  - Zwięzłość + jednoznaczność (GetUserById() zamiast DoUserStuff()).

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

52

Programy najwyższego poziomu

- Możliwość pisania prostych programów bez klasy Program i metody Main().
- Wprowadzono w C# 9, a rozwinięto w nowszych wersjach.
- Kod można pisać bezpośrednio w pliku źródłowym – bardziej zwięzły i czytelny.

Zastosowanie

- Idealne do nauki i przykładów dydaktycznych.
- Przydatne w skryptach, testach i małych narzędziach.
- Ułatwiają szybki start – mniej szablonowego kodu.

Ograniczenia

- W projekcie może być tylko jeden plik z top-level statements.
- W tle kompilator i tak generuje klasę Program i Main().

Gdy projekt rośnie, zwykle i tak przechodzimy do klasycznego układu.

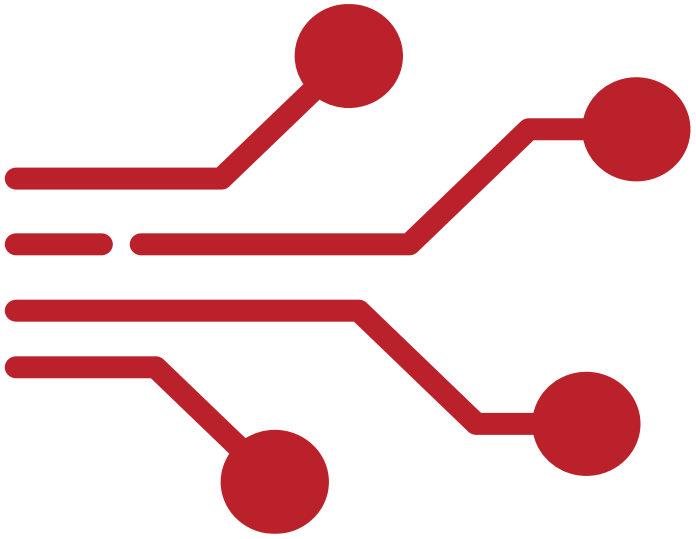
```
Console.WriteLine("Hello, world!");
int x = 5; int y = 7;
Console.WriteLine($"Suma: {x + y}");
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

53

Słownictwo

Czasowniki to zwykle metody



Rzeczowniki to zwykle typy, zmienne, pola i właściwości

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

54

Dyrektywy preprocesora

Służą m.in. do kompilacji warunkowej

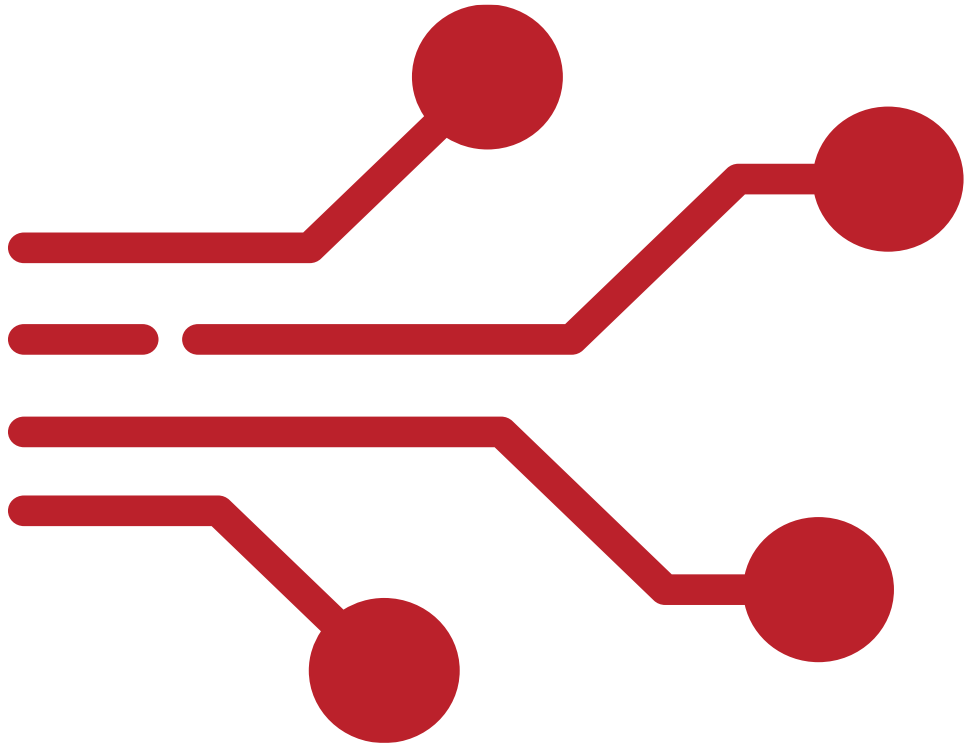
```
static void Main(string[] args)
{
#if DEBUG
        Console.Title = "Kompilacja w trybie \"debug\"";
#endif
        Console.WriteLine("Hello World!");
        Console.ReadLine();
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

55

Zmienne i typy danych



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

56

Deklaracja i zmiana wartości zmiennej

Możemy **zadeklarować** zmienne w dowolnym miejscu metod i klas. Zadeklarowane zmienne nie mogą jednak zostać użyte przed przypisaniem im jakiejś wartości, tj. zanim zostaną **zainicjowane**.

```
int i; // typ całkowity czterobajtowy (32 bity)
long l; // typ całkowity ośmiobajtowy (64 bity)
string s; // łańcuch
float f; // typ rzeczywisty czterobajtowy (32 bity)
double d; // typ rzeczywisty ośmiobajtowy (64 bity)
i = 1; //inicjacja
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

57

Zmienne lokalne i zasięg zmiennych

Zmienne lokalne są deklarowane wewnątrz metod i istnieją wyłącznie w czasie, gdy dana metoda jest wywoływana. Gdy tylko metoda zwróci sterowanie, pamięć zajmowana przez jej zmienne lokalne zostaje zwolniona dla innych zastosowań.

Zasięg:

- Określa, gdzie w kodzie zmienna jest widoczna i dostępna.
- Chroni przed przypadkowym użyciem niewłaściwych danych.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

58

Typy wartościowe i referencyjne

Typy wartościowe

- Przechowywane bezpośrednio na stosie (stack).
- Kopiowanie - tworzy niezależną kopię wartości.
- Przykłady: int, double, bool, struct, enum.

Typy referencyjne

- Przechowywana jest referencja (adres w pamięci).
- Kopiowanie - obie zmienne wskazują na ten sam obiekt w pamięci sterty (heap).
- Przykłady: string, tablice, obiekty.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

59

Przykład

```
int x = 10;
int y = x;   // kopiujemy wartość
y = 20;
Console.WriteLine(x); // 10
Console.WriteLine(y); // 20
string s1 = "Ala";
string s2 = s1;   // kopiujemy referencję
s2 += " ma kota"; // tworzony jest nowy obiekt!
Console.WriteLine(s1); // "Ala"
Console.WriteLine(s2); // "Ala ma kota"
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00
Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

60

Domyślne wartości typów

Każdy typ ma przypisaną wartość domyślną, gdy nie został jawnie zainicjalizowany. Dotyczy pól klas, struktur i tablic – zmienne lokalne muszą być zainicjalizowane.

Przykłady

int -> 0

double -> 0.0

bool -> false

char -> '\0' (znak pusty, null-character)

string -> null

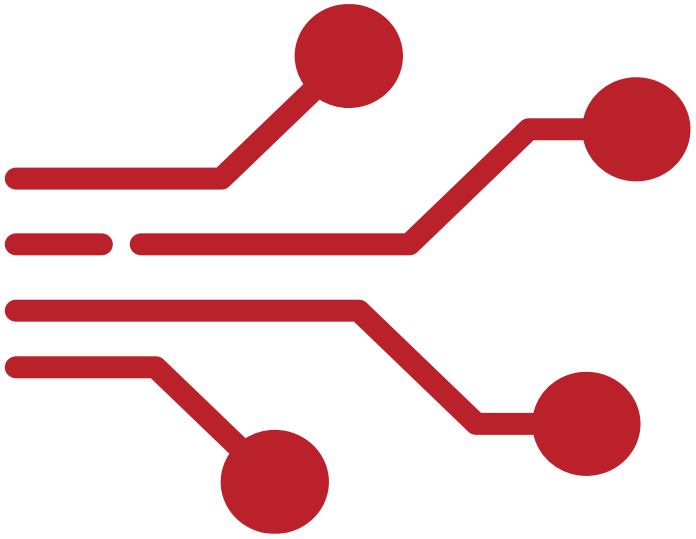
Tablice -> null (dopóki nie są utworzone przez new)

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

61

Sprawdzenie

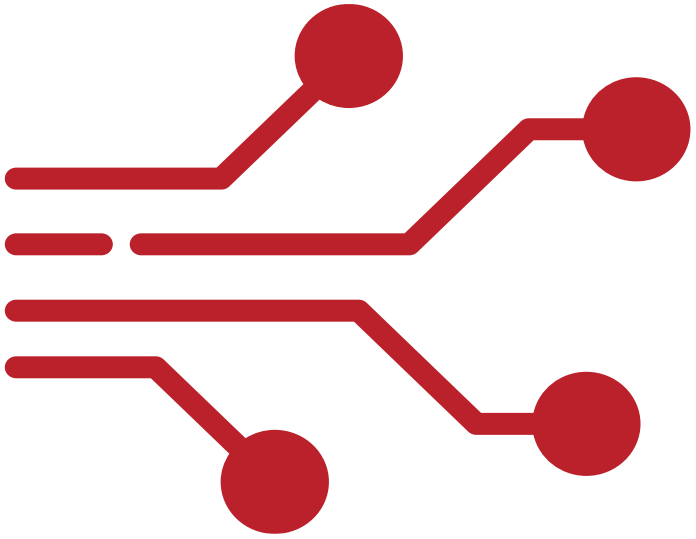


```
int liczba = default;      // 0
bool flaga = default;      // false
string tekst = default;    // null

Console.WriteLine(liczba); // 0
Console.WriteLine(flaga);  // False
Console.WriteLine(tekst);  // nic, bo null
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

62

Określanie typu zmiennej przy inicjalizacji i pseudotyp var



```
var i = 5;
var l = 5L;
var s = "MojaNazwa";
var f = 1.0f;
var d = 1.0;
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

63

null/nullable

Kolejną różnicą między typem wartościowym a referencyjnym jest to, że tylko ten drugi może mieć wartość null. To oznacza, że zmienna typu referencyjnego może nie wskazywać na żaden obiekt. Zmienna wartościowa też może nie być zainicjowana, ale kompilator wychwytuje próby użycia takiej zmiennej i na to nie pozwala.

Czasem możemy jednak chcieć zdefiniować np. liczbę całkowitą typu int mogącą przyjmować dowolne wartości dodatnie, ujemne i zero, ale oprócz tego dopuścić możliwość, że zmienna ta nie jest zainicjowana, bo obliczenia prowadzące do tego jeszcze się nie wykonały lub się nie udały.

int? ni = 1;

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

64

Typ object

Typ object (System.Object) reprezentuje najwyższą klasę stanowiącą podstawę wszystkich typów. Do typu object można rzutować w górę każdy inny typ.

Pakowanie (ang. *boxing*) to proces konwersji egzemplarza typu wartościowego na egzemplarz typu referencyjnego. Typ referencyjny może być klasą object lub interfejsem.

```
int x = 9;

object obj = x; // pakuje int
```

Rozpakowywanie to operacja odwrotna, tzn. polegająca na rzutowaniu obiektu na pierwotny typ wartościowy:

```
int y = (int)obj; // rozpakowanie int
```

Rozpakowywanie można przeprowadzić tylko jako rzutowanie jawne.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

65

Rzutowanie

- Automatyczna konwersja, gdy nie ma ryzyka utraty danych.

```
int x = 10;
double y = x; // int -> double (OK)
```

- Rzutowanie jawne.

Wymaga zapisu w nawiasach, gdy możliwa jest utrata danych.

```
double a = 9.7;
int b = (int)a; // wynik 9
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

66

Rzutowanie

- Operatory as i is

as próbuje rzutować, w razie niepowodzenia zwraca null.

is sprawdza typ przed rzutowaniem.

```
object obj = "Hello";
string? s = obj as string; // "Hello"
if (obj is string txt)     // pattern matching
    Console.WriteLine(txt.ToUpper());
```

- Klasa Convert i metody Parse/TryParse.

```
string liczba = "123";
int n = int.Parse(liczba);        // 123
int m = Convert.ToInt32("456");   // 456
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

67

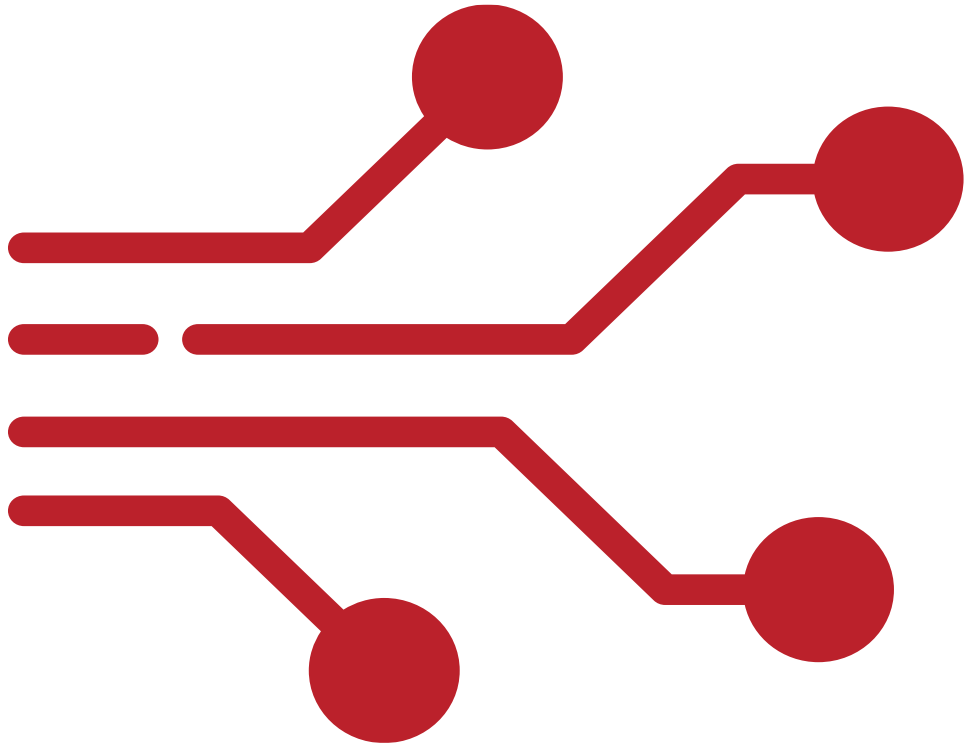
Typ wyliczeniowy

```
public enum MiesiaceLata : byte
{
    Czerwiec = 1,
    Lipiec = 2,
    Sierpien = 3,
    Wrzesien = 4
}
//Domyślnie int
public enum MiesiaceLata
{
    Czerwiec = 1,
    Lipiec = 2,
    Sierpien = 3,
    Wrzesien = 4
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

68

Typy proste i literały



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

69

Proste typy danych

Nazwa typu (słowo kluczowe, alias klasy)	Klasa z obszaru nazw System	Liczba zajmowanych bitów	Możliwe wartości (zakres)	Domyślna wartość
bool	Boolean	1 bajt = 8 bitów	true, false	false
sbyte	SByte	8 bitów ze znakiem	od –128 do 127	0
byte	Byte	8 bitów bez znaku	od 0 do 255	0
short	Int16	2 bajty = 16 bitów ze znakiem	od –32 768 do 32 767	0
int	Int32	4 bajty = 32 bity ze znakiem	od –2 147 483 648 do 2 147 483 647	0
long	Int64	8 bajtów = 64 bity ze znakiem	od –9 223 372 036 854 775 808 do 9 223 372 036 854 775 807	0L
ushort	UInt16	2 bajty = 16 bitów bez znaku	maks. 65 535	0
uint	UInt32	4 bajty = 32 bity bez znaku	maks. 4 294 967 295	0
ulong	UInt64	8 bajtów = 64 bity bez znaku	maks. 18 446 744 073 709 551 615	0UL
char	Char	2 bajty = 16 bitów (zob. komentarz)	Znaki Unicode od U+0000 do U+FFFF (od 0 do 65 535)	'\0'
float	Single	4 bajty (zapis zmiennoprzecinkowy)	Wartość może być dodatnia lub ujemna; najmniejsza bezwzględna wartość to $1{,}4 \cdot 10^{-45}$, największa to ok. $3{,}403 \cdot 10^{38}$	0.0F
double	Double	8 bajtów (zapis zmiennoprzecinkowy)	Najmniejsza wartość to $4{,}9 \cdot 10^{-324}$, największa to ok. $1{,}798 \cdot 10^{308}$	0.0D
decimal	Decimal	16 bajtów (większa precyzja, ale mniejszy zakres niż double)	Najmniejsza wartość to 10^{-28}, największa to $7{,}9 \cdot 10^{28}$	0.0M

Rys. 5. Typy danych [źródło: J. Matulewski, C#. Lekcje programowania. Praktyczna nauka programowania dla platform .NET i .NET Core, Helion 2021]

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Łańcuchy

Funkcja wartość nazwa(argumenty)	Opis	Przykład użycia
indeksator []	Zwraca znak na wskazanej pozycji; pozycja pierwszego znaku to 0	`"Helion"[0]`
bool Equals(*string*)	Porównuje łańcuch z podanym w argumencie	`"Helion".Equals("HELP")`
int IndexOf(*char*), *int* IndexOf(*String*)	Pierwsze położenie litery lub łańcucha; –1, gdy nie zostanie znaleziony	`"Lalalalala".IndexOf('a')`
int LastIndexOf(*char*), *int* LastIndexOf(*String*)	Jak wyżej, ale ostatnie wystąpienie litery lub łańcucha	`"Lalalalala".LastIndexOf('a')`
int Length	Zwraca długość łańcucha, właściwość	`"Helion".Length`
string Replace(*string*,*string*), *string* Replace(*char*,*char*)	Zamienia wskazany fragment na inny	`"HELP".Replace("P","ION")`
string Substring(*int*,*int*)	Fragment łańcucha od pozycji w pierwszym argumencie o długości podanej w drugim	`"Wydawnictwo".Substring(5,3)`
string Remove(*int*,*int*)	Usuwa wskazany fragment	`"Helion".Remove(1,2)`
string Insert(*int*,*string*)	Wstawia łańcuch przed literą o podanej pozycji	`"Helion".Insert(2, "123")`
string ToLower() *string* ToUpper()	Zmienia wielkość wszystkich liter	`"Helion".ToLower()`
string Trim() oraz TrimStart, TrimEnd	Usuwa spacje z przodu i z tyłu łańcucha	`"  Helion  ".Trim()`
string PadLeft(*int*,*char*) *string* PadRight(*int*,*char*)	Uzupełnia łańcuch znakiem podanym w drugim argumencie, aż do osiągnięcia długości zadanej w pierwszym argumencie	`"Helion".PadLeft(10,' ')`
bool EndsWith(*string*), *bool* StartsWith(*string*)	Sprawdza, czy łańcuch rozpoczyna się lub kończy podanym fragmentem	`"csc.exe".EndsWith("exe")`

Rys. 6. Typy danych [źródło: J. Matulewski, C#. Lekcje programowania. Praktyczna nauka programowania dla platform .NET i .NET Core, Helion 2021]

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Możliwości przechowywania tekstu

- **Literały ciągów znaków** — znaki zamknięte w cudzysłowie. Można w nich stosować znaki modyfikacji, takie jak \t oznaczający tabulację. Aby zapisać lewy ukośnik, piszemy \\.
- **Surowe literały ciągów znaków** — znaki zamknięte pomiędzy przynajmniej trzema znakami cudzysłowu.
- **Dosłowne ciągi znaków** — literał ciągu znaków poprzedzony znakiem @, który powoduje wyłączenie interpretowania znaków modyfikacji. W tym przypadku lewy ukośnik jest po prostu lewym ukośnikiem. Umożliwiają też rozciągnięcie wartości typu string na kilka wierszy, ponieważ znaki końca wiersza traktowane są jak znaki końca wiersza w tekście, a nie jak instrukcje dla kompilatora.
- **Interpolowany ciąg znaków** — literał ciągu znaków poprzedzony znakiem $, co pozwala wbudowywać formatowane zmienne.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

72

Możliwości przechowywania tekstu. Przykłady

```
string tekst = "Ala\tma\tkota";  // tabulatory
string sciezka = "C:\\Program Files\\App"; // zapis ukośnika

string html = """
<html>
  <body>
    <h1>Hello!</h1>
  </body>
</html>
""";
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

73

Możliwości przechowywania tekstu. Przykłady

```
string sciezka = @"C:\Users\Student\Dokumenty";
string wiersze = @"Pierwsza linia
Druga linia
Trzecia linia";
```

```
int a = 5, b = 7;
string wynik = $"Suma {a} + {b} = {a + b}";
string sciezka = @$"C:\Uzytkownicy\{Environment.UserName}\Pulpit";
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

74

Wybrane metody klasy string

Funkcja wartość nazwa(argumenty)	Opis	Przykład użycia
indeksator []	Zwraca znak na wskazanej pozycji; pozycja pierwszego znaku to 0	`"Helion"[0]`
bool Equals(*string*)	Porównuje łańcuch z podanym w argumencie	`"Helion".Equals("HELP")`
int IndexOf(*char*), *int* IndexOf(*String*)	Pierwsze położenie litery lub łańcucha; –1, gdy nie zostanie znaleziony	`"Lalalalala".IndexOf('a')`
int LastIndexOf(*char*), *int* LastIndexOf(*String*)	Jak wyżej, ale ostatnie wystąpienie litery lub łańcucha	`"Lalalalala".LastIndexOf('a')`
int Length	Zwraca długość łańcucha, właściwość	`"Helion".Length`
string Replace(*string*,*string*), *string* Replace(*char*,*char*)	Zamienia wskazany fragment na inny	`"HELP".Replace("P","ION")`
string Substring(*int*,*int*)	Fragment łańcucha od pozycji w pierwszym argumencie o długości podanej w drugim	`"Wydawnictwo".Substring(5,3)`
string Remove(*int*,*int*)	Usuwa wskazany fragment	`"Helion".Remove(1,2)`
string Insert(*int*,*string*)	Wstawia łańcuch przed literą o podanej pozycji	`"Helion".Insert(2, "123")`
string ToLower() *string* ToUpper()	Zmienia wielkość wszystkich liter	`"Helion".ToLower()`
string Trim() oraz TrimStart, TrimEnd	Usuwa spacje z przodu i z tyłu łańcucha	`"  Helion  ".Trim()`
string PadLeft(*int*,*char*) *string* PadRight(*int*,*char*)	Uzupełnia łańcuch znakiem podanym w drugim argumencie, aż do osiągnięcia długości zadanej w pierwszym argumencie	`"Helion".PadLeft(10,' ')`
bool EndsWith(*string*), *bool* StartsWith(*string*)	Sprawdza, czy łańcuch rozpoczyna się lub kończy podanym fragmentem	`"csc.exe".EndsWith("exe")`

Rys. 7 Działania na stringach [źródło: J. Matulewski, C#. Lekcje programowania. Praktyczna nauka programowania dla platform .NET i .NET Core, Helion 2021]

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

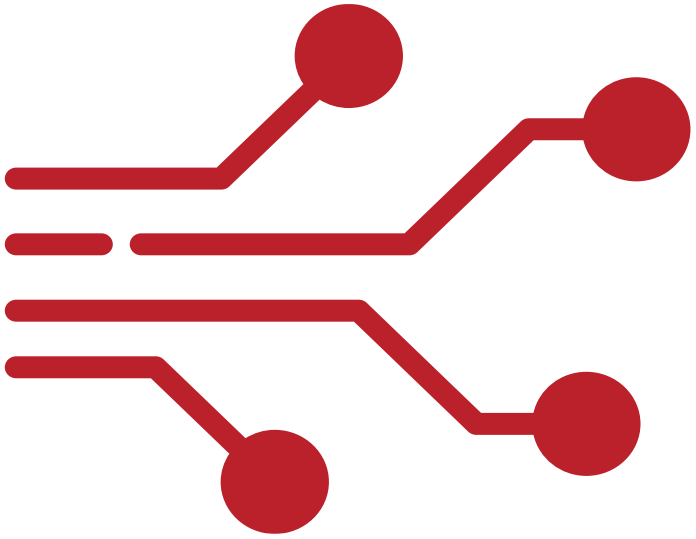
String a StringBuilder

```
//string
string s = "abc---";
s += "xyz";
s = s.Replace("---", " ijk ");
Console.WriteLine(s);
//StringBuilder
System.Text.StringBuilder sb = new
System.Text.StringBuilder("abc---");
sb.Append("xyz");
sb.Replace("---", " ijk ");
Console.WriteLine(sb.ToString());
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Wartości logiczne

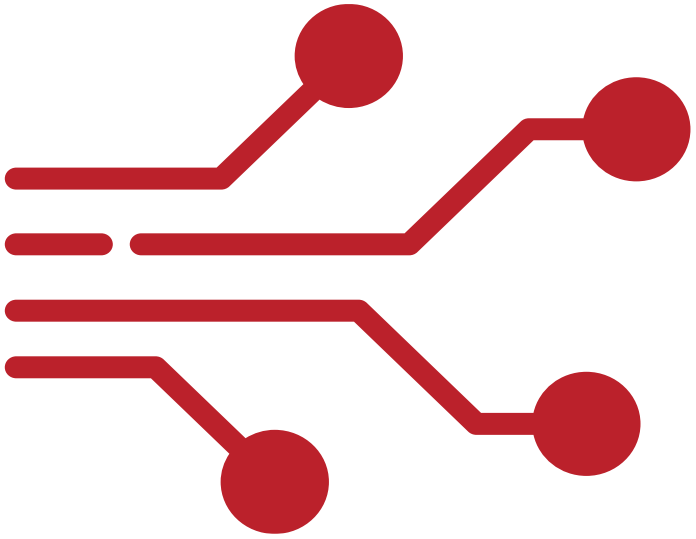
```
bool ok = true;
bool notOk = false;
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

77

Konkatenacja łańcuchów i ich podział



```
string a = "Ala";
string b = " ma kota";
string c = a + b;                    // "Ala ma kota"
string d = string.Concat(a, b);      // "Ala ma kota"

string tekst = "Ala ma kota";
string[] slowa = tekst.Split(' ');
// slowa = ["Ala", "ma", "kota"]
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

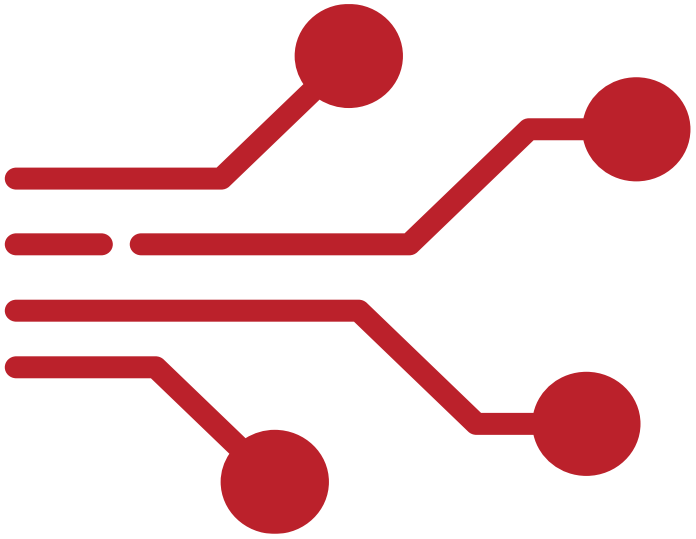
78

Operatory arytmetyczne

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

79

Operatory. Inkrementacja i dekrementacja

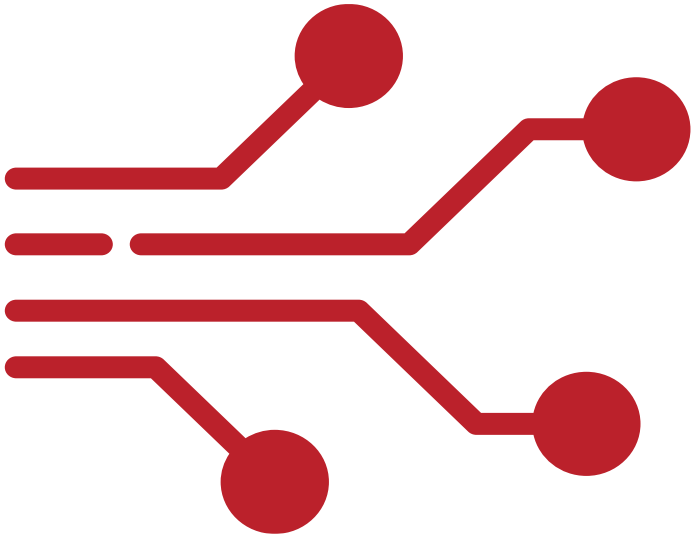


```
int i = 1;
i = i + 2;
Console.WriteLine(i);
int i = 1;
i += 2;
Console.WriteLine(i);
//Analogicznie w przypadku --
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

80

Dzielenie całkowite i modulo



```
int a = 7, b = 2;
Console.WriteLine(a / b); // 3
double x = 7, y = 2;
Console.WriteLine(x / y); // 3,5
int a = 7, b = 2;
Console.WriteLine(a % b); // 1
int n = 10;
if (n % 2 == 0)
    Console.WriteLine("Parzysta");
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

81

Operatory predefiniowane

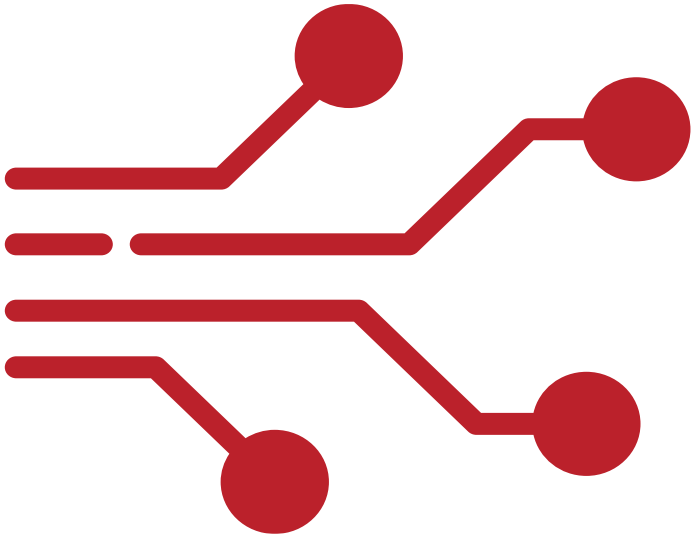
Grupa operatorów	Operator	Opis
Podstawowe (ang. *primary*)	`x.y`, `x?.y`	Dostęp do pola, metody, właściwości i zdarzenia. Operator `?.` zwraca `null`, jeżeli `x` jest `null`.
	`f()`, `f(x)`	Wywołanie metody
	`a[n]`, `a?[n]`	Odwołanie do elementu tablicy lub indeksatora. Operator `?[]` zwraca `null`, jeżeli `a` jest `null`. Argumentem `[]` może być indeks lub zakres indeksów.
	`x++`, `x--`	Zwiększenie lub zmniejszenie wartości o 1 **po** wykonaniu innej instrukcji (np. po wykonaniu `int x=2; int y=x++;` otrzymamy `y=2, x=3`)
	`new`	Tworzenie obiektu, składnia: `Klasa referencja=new Klasa(arg_konstr);`
	`default(typ)`	Zwraca domyślną wartość dla podanego typu. Dla typów referencyjnych to `null`, dla liczb to `0`
	`typeof`	Zwraca zmienną typu `System.Type` opisującą typ argumentu: `typeof(int).ToString()` równy jest `System.Int32`
	`sizeof(typ)`	Zwraca liczbę bajtów zajmowanych przez zmienną; `sizeof(float)` to 4, a `sizeof(double)` to 8
	`nameof(zmienna)`	Zwraca łańcuch (`string`) z nazwą zmiennej
	`checked, unchecked`	Operatory kontrolujące zgłaszanie wyjątku przekroczenia zakresu dla operacji na liczbach całkowitych `int i = unchecked(int.MaxValue + 1);`
Jednoargumentowe (ang. *unary*)	`+x, -x`	Operatory zmiany znaku liczby (np. –3)
	`!`	Negacja logiczna (np. `!true` to `false`)

Rys. 8 Operatory predefiniowane
[źródło: J. Matulewski, C#. Lekcje programowania. Praktyczna nauka programowania dla platform .NET i .NET Core, Helion 2021]

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

82

Operatory predefiniowane



Rys. 9 Operatory predefiniowane
[źródło: J. Matulewski, C#. Lekcje programowania. Praktycz
nauka programowania dla platform .NET i .NET Core, Helior
2021]

Grupa operatorów	Operator	Opis
	`~`	Negacja bitowa (zdefiniowana dla typów `int`, `uint`, `long` i `ulong`): `~01001010 = 10110101`
	`++x`, `--x`	Zwiększenie lub zmniejszenie wartości o 1 **przed** wykonaniem innej instrukcji (np. w wyniku wykonania operacji `int x=2; int y=++x;` otrzymamy `y=3`, `x=3`)
	`(typ)x`	Jawne rzutowanie (konwersja) zmiennej `x` na typ wskazany w nawiasie. Uwaga! `(int)0.99 = 0`
	`await`	Wstrzymuje wykonywanie metody asynchronicznej, w której jest umieszczony, do zakończenia operacji umieszczonej za tym operatorem (zob. dodatek)
Zakresu	`..`	`a[2..5]` to tablica o elementach `a[2]`, `a[3]` i `a[4]`
Mnożenia (ang. *multiplicative*)	`*`, `/`, `%`	Operator dzielenia zwraca wynik zgodny z typem argumentów: `5/2=2`, `5/2.0=2.5`, `5f/2=2.5`
		Operator reszty z dzielenia: `5%2=1`
Dodawania (ang. *additive*)	`+`, `-`	Dodawanie i odejmowanie. Typ wyniku zależy od typu argumentów, więc: `uint i=1; uint j=3; uint k=i-j;` spowoduje, że `k=4294967294`
Przesunięcia bitów (ang. *shift*)	`<<`, `>>`	`1 << 1 = 2`, bo jedynka została przesunięta z ostatniej pozycji na przedostatnią i zostało dostawione zero
Porównania wartości i typów (ang. *relational and type testing*)	`<`, `>`, `<=`, `>=`	Porównanie wartości: `1>2` ma wartość `false`
	`is`	Porównanie typów: `1 is int` ma wartość `true`
	`as`	Rzutowanie równoznaczne z: *`wyrażenie`* `is` *`typ`* `?` `(typ)`*`wyrażenie`* `:` `(typ)null`
Równości (ang. *equality*)	`==`, `!=`	Wyrażenie `0.5 == 1/2f` jest prawdziwe (wartość logiczna), `0.5!=1/2`, czyli `0.5!=0` jest również prawdziwe
Operacje na bitach	`&`	Bitowe `AND` (i), tzn. bit jest zapalany tylko wtedy, jeżeli oba odpowiednie bity argumentów są zapalone
		`74&15=10` (`01001010 & 00001111 = 00001010`)
	`^`	Bitowe `XOR` (rozłączne lub) — bit jest równy 1 tylko wtedy, gdy odpowiednie bity argumentów są różne
		`74^15=69` (`01001010 ^ 00001111 = 01000101`)
	`|`	Bitowe `OR` (lub) — bit jest zapalony, jeżeli przynajmniej jeden odpowiedni bit argumentów jest zapalony
		`74|15=79` (`01001010 | 00001111 = 01001111`)
Operacje na wartościach logicznych (`bool`)	`&&`	Logiczne `AND` (`true` jedynie wtedy, gdy oba argumenty są `true`)
	`||`	Logiczne `OR` (`true`, gdy przynajmniej jeden argument jest `true`)

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Operatory predefiniowane

Grupa operatorów	Operator	Opis
Warunkowy (ang. *conditional*)	`?:`	*warunek*`?`*wartość_jeżeli_true*`:`*wartość_jeżeli_false*, gdzie *warunek* musi mieć wartość typu `bool`, np. `x>y?x:y` zwraca większą wartość spośród `x` i `y`

Przypisania (ang. *assignment*)	`=` oraz `*=`, `/=`, `%=`, `+=`, `-=`, `<<=`, `>>=`, `&=`, `^=`, `|=`	Inicjacja i zmiana wartości zmiennych, np. polecenia:
		`int x=1;`
		`x+=2;`
		nadadzą zmiennej `x` wartość 3

Deklaracja lambda	`??`	`int j = i ?? -1;` Jeżeli `i` jest równe `null`, to wartością wyrażenia `i ?? -1` jest -1, w przeciwnym wypadku wartością jest wartość `i`

Deklaracja lambda Nowa forma deklarowania metod	`=>`	Definiowanie wyrażeń lambda (por. rozdział 4.):
		`(int x) => { return x * x; }`
		Definiowanie metod:
		`int kwadrat(int arg) => arg * arg;`

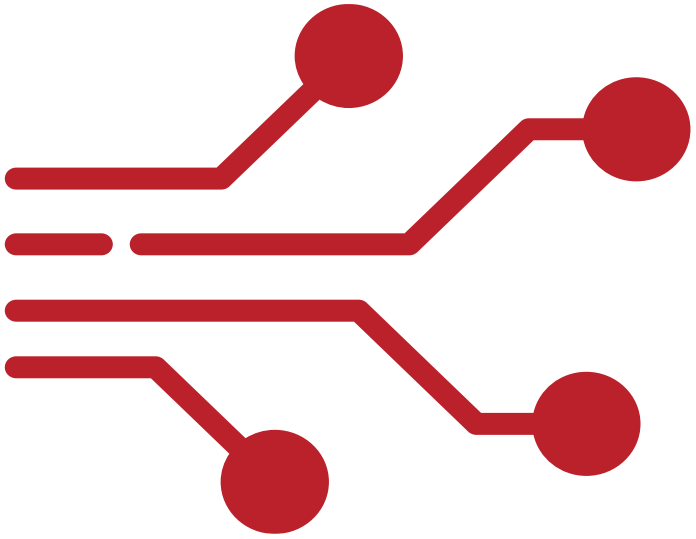
Rys. 10 Operatory predefiniowane
[źródło: J. Matulewski, C#. Lekcje programowania. Praktyczna nauka programowania dla platform .NET i .NET Core, Helion 2021]

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

84

Kolejność

wykonywania



Rys. 11 Kolejność wykonywania [źródło: J. Albahari, C# 12 w pigułce, Helion 2024]

Dostępu do składowej	`x.y`
Warunkowy z kontrolą wartości null	`x?.y` lub `x?[0]`
Akceptujący null	`x!*y` lub `x![0]`
Wskaźnik do struktury	`x->y`
Wywołania funkcji	`x()`
Tablicowy lub indeksowania	`a[x]`
Postinkrementacja	`x++`
Postdekrementacja	`x--`
Tworzenia egzemplarza	`new Foo()`
Alokacji na stosie	`stackalloc(10)`
Sprawdzania typu po identyfikatorze	`typeof(int)`
Sprawdzania nazwy identyfikatora	`nameof(x)`
Włączania kontroli przepełnienia całkowitoliczbowego	`checked(x)`
Wyłączania kontroli przepełnienia całkowitoliczbowego	`unchecked(x)`
Wartość domyślna	`default(char)`
Oczekiwanie	`await myTask`
Sprawdzanie rozmiaru struktury	`sizeof(int)`
Wartość dodatnia	`+x`
Wartość ujemna	`-x`
Negacja	`!x`
Dopełnienie bitowe	`~x`
Preinkrementacja	`++x`
Predekrementacja	`--x`
Rzutowanie	`(int)x`
Indeksowanie od końca	`array[^1]`
Wartość pod adresem	`*x`
Adres wartości	`&x`
Zakresu indeksów	`x..y` `x..^y`

Wyrażenie switch	`num switch {`
	`  1 => true,`
	`  _ => false`
	`}`

Wyrażenie with	`rec with` `{ X = 123 }`

Mnożenie	`x * y`
Dzielenie	`x / y`
Reszta z dzielenia	`x % y`
Dodawanie	`x + y`
Odejmowanie	`x - y`
Przesunięcie w lewo	`x << 1`
Przesunięcie w prawo	`x >> 1`
Przesunięcie w prawo bez znaku	`x >>> 1`
Mniejszość	`x < y`
Większość	`x > y`
Mniejszy lub równy	`x <= y`
Większy lub równy	`x >= y`
Typ jest czymś lub podklasą czegoś	`x is y`
Konwersja typów	`x as y`
Jest równy	`x == y`
Nie jest równy	`x != y`
I	`x & y`
Lub wykluczające	`x ^ y`
Lub	`x | y`
Warunkowe i	`x && y`
Warunkowe lub	`x || y`
Sprawdzanie null	`x ?? y`
Warunkowy	`jeśliPrawda ? taWartoś` `: wPrzeciwnymRazieTa`

Przypisanie	`x = y`
Mnożenie zmiennej przez wartość	`x *= 2`
Dzielenie zmiennej przez wartość	`x /= 2`
Reszta i przypisanie do siebie	`x %= 2`
Dodanie do zmiennej	`x += 2`
Odjęcie od zmiennej	`x -= 2`
Przesunięcie o wartość w lewo	`x <<= 2`
Przesunięcie o wartość w prawo	`x >>= 2`
Przesunięcie w prawo bez znaku z przypisaniem do siebie	`x >>>= 2`
I logiczne przez	`x &= 2`
Lub wykluczające przez	`x ^= 2`
Lub przez	`x |= 2`
Przypisanie łączące z kontrolą wartości null	`x ??= 0`
Lambda	`x => x + 1`

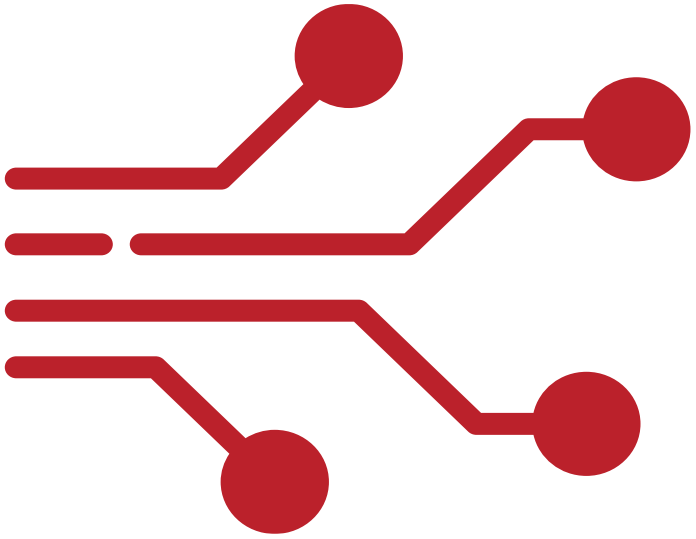
LITECHNIKA BELSKA

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

ofinansowane przez Unię Europejską

Operatory porównania

==, !=, <, >, <=, >=

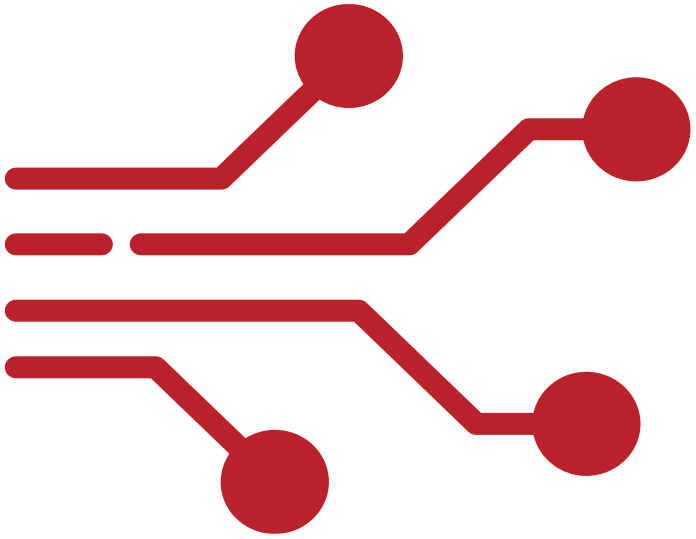


`Equals(…)`

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

86

Konwersje typów

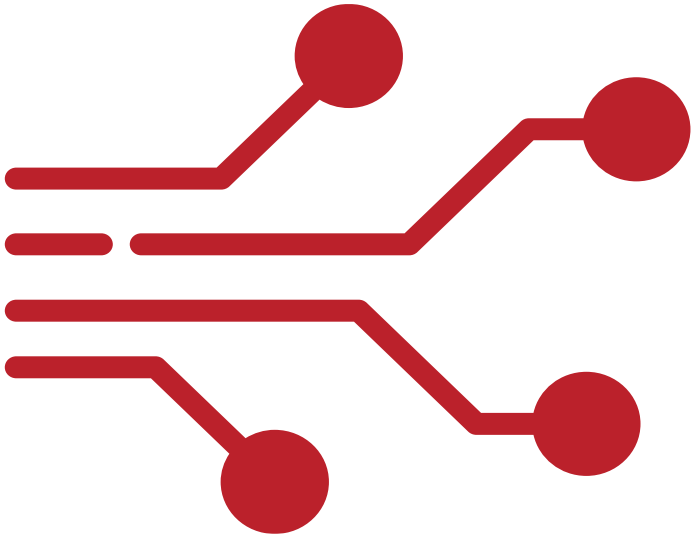


```
double x = 1; // to jest OK
int i = 1.0; // tu pojawi się błąd
1.0 + 1 // double
1.0 / 2 // double, wartość 0.5
1 / 2 // int, wartość 0
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

87

Parsowanie stringów



```
string liczbaTxt = "123";
int liczba = int.Parse(liczbaTxt);          // 123
double d = double.Parse("3,14",
    new CultureInfo("pl-PL"));              // 3,14
bool ok = int.TryParse("abc", out int wynik);
// ok = false, wynik = 0
string dataTxt = "01.09.2025";
DateTime data = DateTime.Parse(dataTxt,
    new CultureInfo("pl-PL"));
Console.WriteLine(data); // 01.09.2025 00:00:00
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

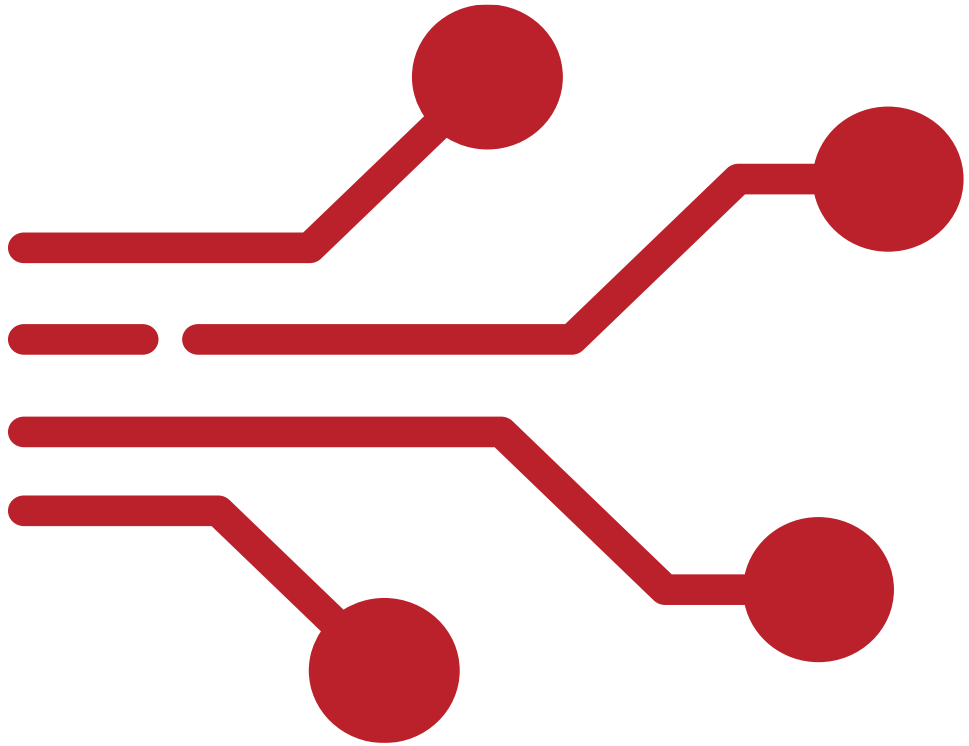
88

Daty

```
DateTime data = new DateTime(2025, 9, 1);
Console.WriteLine(data.ToString("d", new CultureInfo("pl-PL")));
// 01.09.2025
Console.WriteLine(data.ToString("d", new CultureInfo("en-US")));
// 9/1/2025
Console.WriteLine(data.ToString("d", new CultureInfo("en-GB")));
// 01/09/2025
Console.WriteLine(data.ToString("yyyy-MM-dd",
    CultureInfo.InvariantCulture));
// 2025-09-01
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Operatory logiczne i warunkowe



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

90

Operatory logiczne

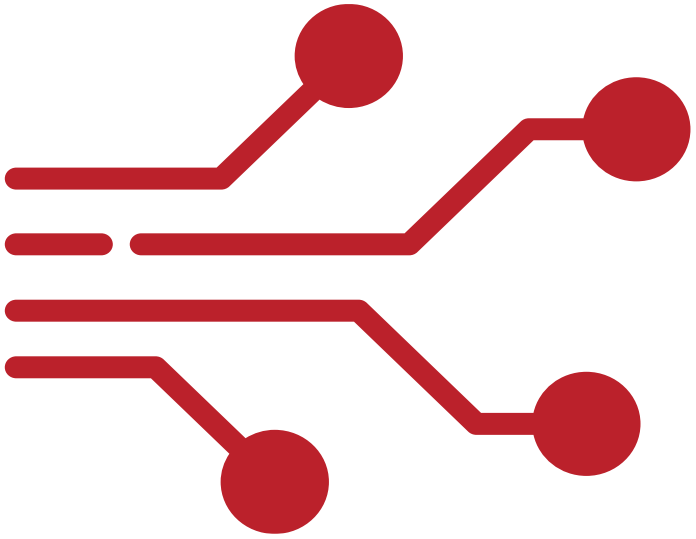
- AND && – zwraca true, jeśli oba warunki są prawdziwe.
- OR || – zwraca true, jeśli choć jeden warunek jest prawdziwy.
- NOT ! – odwraca wartość logiczną.

```
bool a = true;
bool b = false;
Console.WriteLine(a && b); // false
Console.WriteLine(a || b); // true
Console.WriteLine(!a);     // false
int wiek = 25;
bool student = true;
if (wiek >= 23 && student)
    Console.WriteLine("Zniżka!");
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

91

Operator trójargumentowy

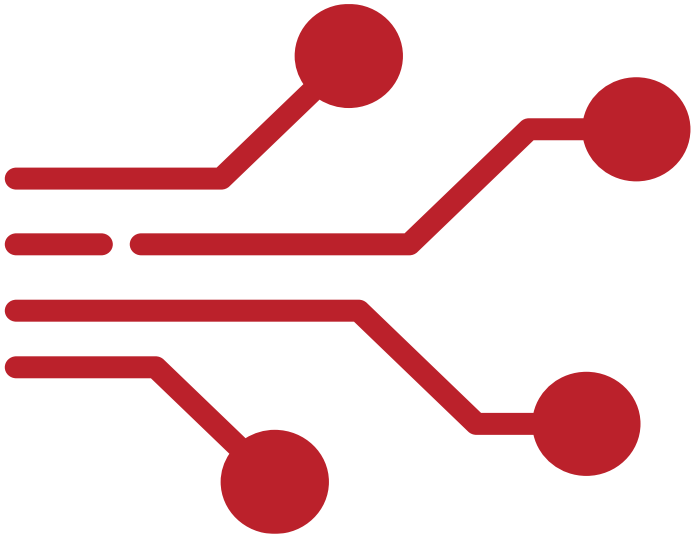


```
// Składnia operatora warunkowego
var wynik = wyrażenie_logiczne ?
wartość_jeśli_prawda : wartość_jeśli_fałsz;
// Przykład operatora warunkowego
string wynik = x > 5 ? "Większe niż 5" :
"Mniejsze niż lub równe 5";
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

92

Operator pomijania wartości null



```
string? nazwisko = ReadLine(); // Poproś
      //użytkownika o podanie nazwiska

      // W zmiennej maksDl zapisz długość  tekstu
      //ze zmiennej nazwisko,

      // a jeżeli ma ona wartość null,

      //to zapisz wartość 30
int maksDl = nazwisko?.Length ?? 30;

// Jeżeli zmienna nazwisko ma wartość null,
// to otrzyma wartość "nieznany"

nazwiskoAutora ??= "nieznany";
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

93

Operatory bitowe

- AND & – bitowy iloczyn (1, gdy oba bity = 1).
- OR | – bitowe „lub” (1, gdy przynajmniej jeden bit = 1).
- XOR ^ – bitowe „różne” (1, gdy bity są różne).
- NOT ~ – negacja bitowa (odwraca wszystkie bity).

```
int a = 6;   // binarnie  110
int b = 3;   // binarnie  011
Console.WriteLine(a & b); // 2  (010)
Console.WriteLine(a | b); // 7  (111)
Console.WriteLine(a ^ b); // 5  (101)
Console.WriteLine(~a);    // -7 (w kodzie U2)
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

94

Operatory przesunięć

- << – przesunięcie bitów w lewo (mnożenie przez 2^n).
- >> – przesunięcie bitów w prawo (dzielenie całkowite przez 2^n).

```
int x = 3;          // binarnie  0000 0011
Console.WriteLine(x << 1); // 6   (0000 0110)
Console.WriteLine(x << 2); // 12  (0000 1100)
int y = 16;         // binarnie 0001 0000
Console.WriteLine(y >> 1); // 8   (0000 1000)
Console.WriteLine(y >> 2); // 4   (0000 0100)
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

95

Dziękuję za uwagę

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Wykład nr 3
Temat: Instrukcje sterujące. Podstawowe operacje wejścia/wyjścia. Debugowanie

Paweł Karczmarek
Katedra Inteligencji Obliczeniowej Politechniki Lubelskiej

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Plan prezentacji

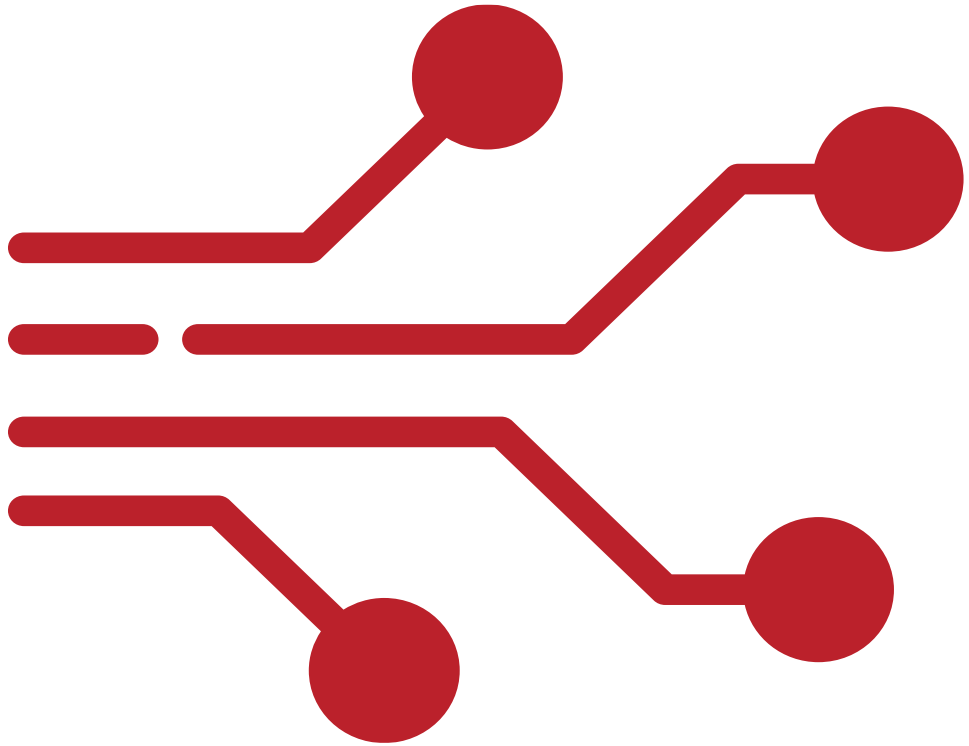
- Instrukcje sterujące
- Obsługa wyjątków
- We/wy
- Debugowanie

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

98

Instrukcje sterujące

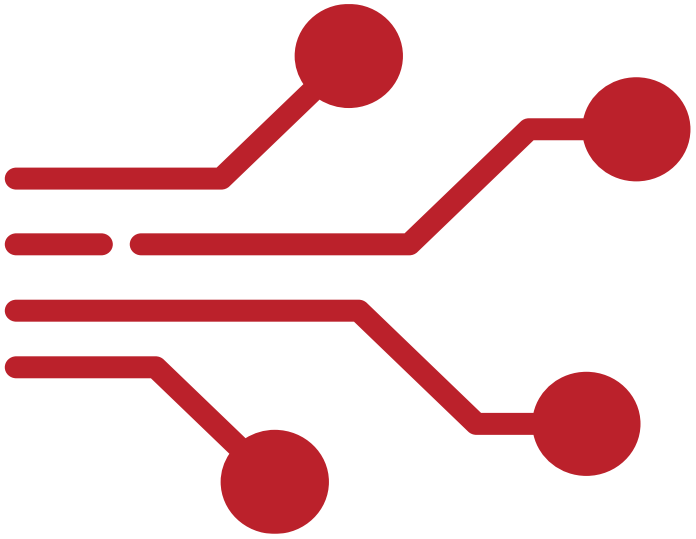


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

99

Instrukcja if

- sprawdza wartość wyrażenia logicznego i na tej podstawie określa, którą ścieżkę kodu należy wykonać. Jeżeli to wyrażenie jest prawdziwe, to wykonywany jest blok kodu. Blok else jest tutaj elementem opcjonalnym i jest wykonywany w przypadku, gdy wyrażenie jest fałszywe. Instrukcje if można w sobie zagnieżdżać.



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Instrukcja if

```
if (wyrażenie1)
{
        // Blok jest wykonywany, jeżeli wyrażenie1 jest prawdziwe
}
else if (wyrażenie2)
{
        // Blok jest wykonywany, jeżeli wyrażenie1 jest fałszywe, ale wyrażenie2
        //jest prawdziwe
}
//itd…
else
{
        // Blok jest wykonywany, jeżeli wszystkie wyrażenia są fałszywe.
}
```

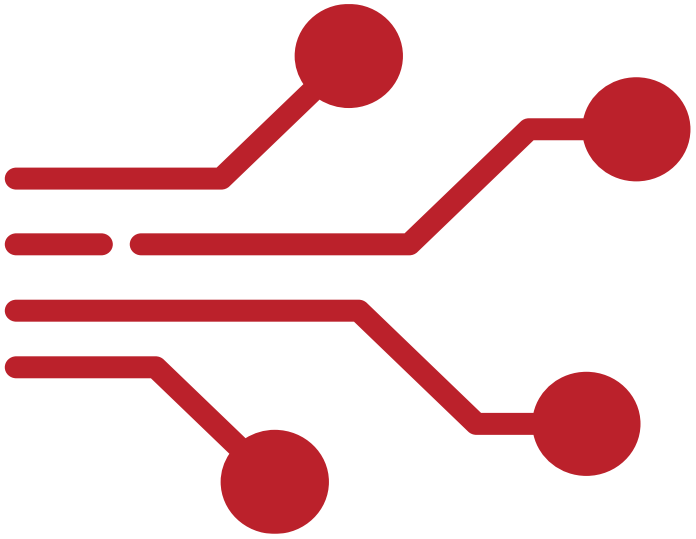

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

101

Przykład

```
string haslo = "mojeSuperHaslo125$";
if (haslo.Length < 8)
{
        WriteLine("Twoje hasło jest za krótkie. Musi mieć
        przynajmniej 8 znaków.");
}
else
{
        WriteLine("Masz silne hasło.");
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

102

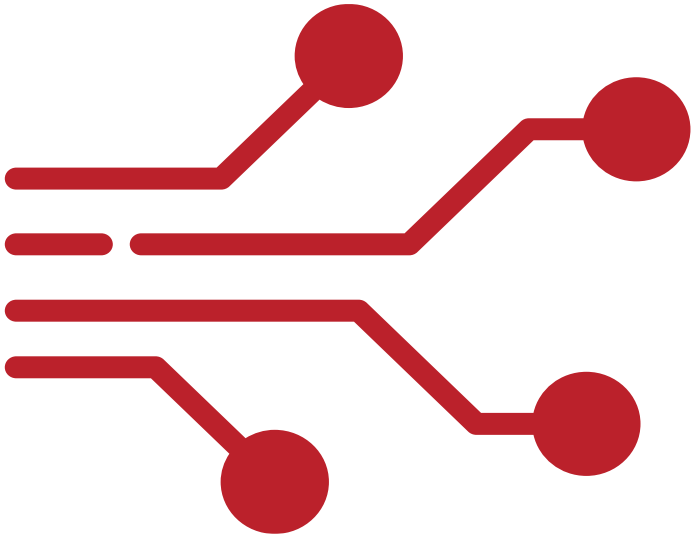
Dopasowywanie wzorców z instrukcją if

```
// Dodanie lub usunięcie cudzysłowu zmieni zachowanie
// tego kodu
object o = "3";
int j = 4;
if(o is int i)
{
        WriteLine($"{i} x {j} = {i * j}");
}
else
{
        WriteLine("o nie ma typu int, a zatem nie mogę wykonać
        mnożenia!");
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

103



Instrukcja switch

porównuje pojedyncze wyrażenie z listą wielu możliwych przypadków. Każdy z takich przypadków powiązany jest z określonym wyrażeniem. Każdy przypadek musi być zakończony:

- słowem kluczowym break
- albo słowami kluczowymi goto case
- albo nie powinien zawierać żadnych instrukcji
- albo słowem kluczowym goto odwołującym się do etykiety
- albo słowem kluczowym return powodującym zakończenie funkcji

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

104

Przykład

```
var liczba = (new Random()).Next(minValue: 1, maxValue: 7);
WriteLine($"Wylosowano liczbę {liczba}");
switch (liczba)
{
        case 1:    WriteLine("Jeden");
                   break;
        case 2:    WriteLine("Dwa");
                   goto case 1;
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

105

Przykład

```
case 3: // Sekcja obsługująca wiele przypadków
      case 4:     WriteLine("Trzy lub cztery");
            goto case 1;
      case 5:     goto Etykieta;
            default:
            WriteLine("Domyślnie");
            break;
} // Koniec instrukcji switch
WriteLine("Koniec instrukcji switch");
Etykieta:
WriteLine("Już za etykietą");
```

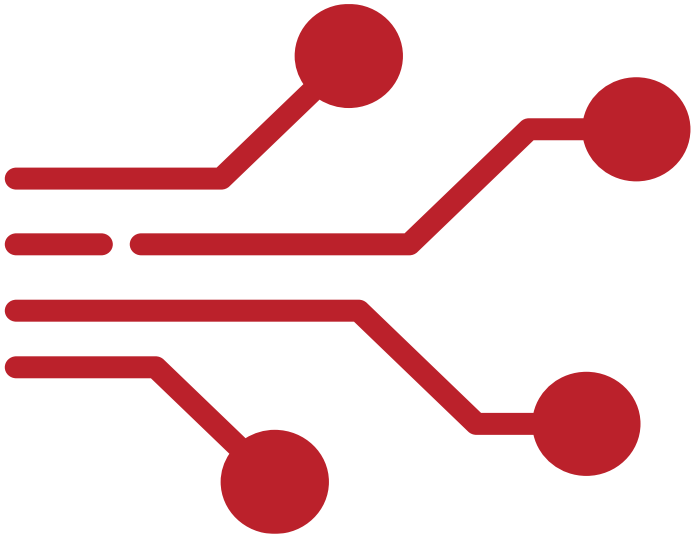

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

106

Klasyczne zastosowania switch

- Przypadek typu wyliczeniowego
- Menu konsoli (w ramach pętli)



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

107

Instrukcja while

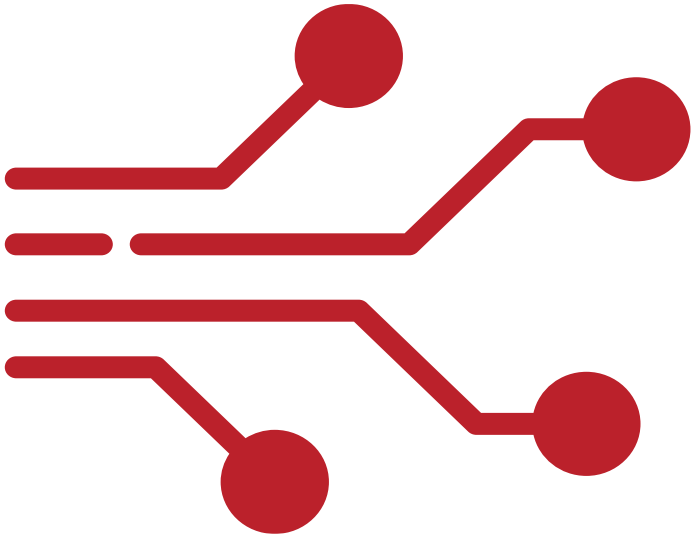
Instrukcja while sprawdza wartość wyrażenia warunkowego i jeżeli jest to wartość true, kontynuuje wykonywanie pętli.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

108

Przykład



```
int x = 0;
while (x < 10)
{
    WriteLine(x);
    x++;
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

109

Instrukcja do

Instrukcja do działa tak samo, jak instrukcja while, z tym że wyrażenie logiczne sprawdzane jest na końcu bloku kodu, a nie na jego początku. Oznacza to, że taki blok zostanie wykonany przynajmniej raz.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

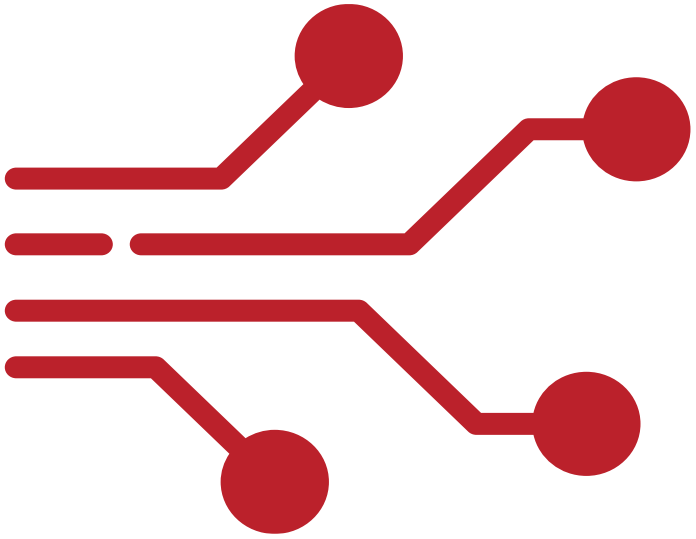
110

Instrukcja do. Przykład

```
string? faktyczneHaslo = "Ha$l0";
string? haslo;
do
{
    Write("Podaj swoje hasło: ");
    haslo = ReadLine();
} while (haslo != faktyczneHaslo);
WriteLine("Tak jest!");
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską



Instrukcja for

Instrukcja for działa podobnie do instrukcji while, przy czym jest znacznie bardziej dokładna. Łączy w sobie:

- **Wyrażenie inicjujące**, które jest wykonywana raz, na samym początku pętli.
- **Wyrażenie logiczne** sprawdzane przed uruchomieniem każdego obiegu pętli, aby sprawdzić, czy ten obieg ma zostać wykonany. Jeżeli wyrażenie zwróci wartość true albo w ogóle go nie ma, to pętla zostanie wykonana ponownie.
- **Wyrażenie iteratora**, które jest wykonywane po każdym obiegu pętli. Tutaj często wykonywana jest inkrementacja zmiennej licznika.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

112

Instrukcja for. Przykłady

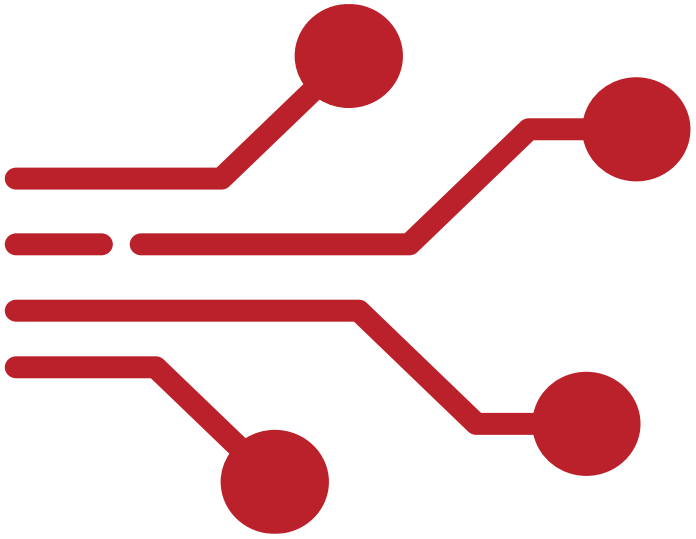
```
for (int y = 1; y <= 10; y++)
{
    WriteLine(y);
}

for (int y = 0; y <= 10; y += 3)
{
    WriteLine(y);
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

113

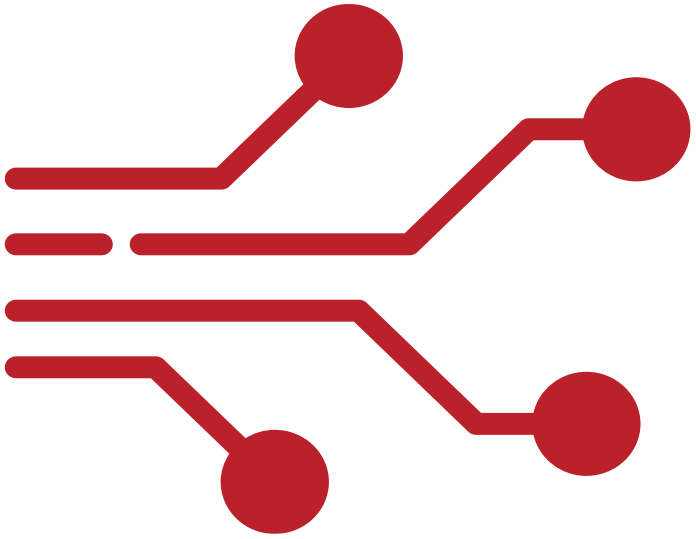
Instrukcja foreach



Używana jest do wykonywania bloku kodu dla każdego elementu sekwencji, takiego jak tablica lub kolekcja. Każdy taki element można jedynie odczytywać, a jakakolwiek modyfikacja sekwencji w trakcie trwania iteracji (np. dodanie lub usunięcie elementu) powoduje zwrócenie wyjątku.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Instrukcja foreach. Przykład



```
string[] imiona = { "Maria", "Bartek",
"Dariusz" };
foreach (string imie in imiona)
{
    WriteLine($"{imie} ma {imie.Length}
    znaków.");
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

115

Przykład

```
Console.Write("Podaj n: ");
int n = int.Parse(Console.ReadLine());
int suma1 = 0;
int i = 1;
while (i <= n)
{
    suma1 += i;
    i++;
}
Console.WriteLine($"Suma (while): {suma1}");
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

116

Przykład

```
Console.Write("Podaj n: ");
int n = int.Parse(Console.ReadLine());
int suma2 = 0;
int j = 1;
do
{
    suma2 += j;
    j++;
}
while (j <= n);
Console.WriteLine($"Suma (do…while): {suma2}");
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00
Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

117

Przykład

```
Console.Write("Podaj n: ");
int n = int.Parse(Console.ReadLine());
int suma3 = 0;


for (int k = 1; k <= n; k++)
{
    suma3 += k;
}
Console.WriteLine($"Suma (for): {suma3}");
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

118

Przykład

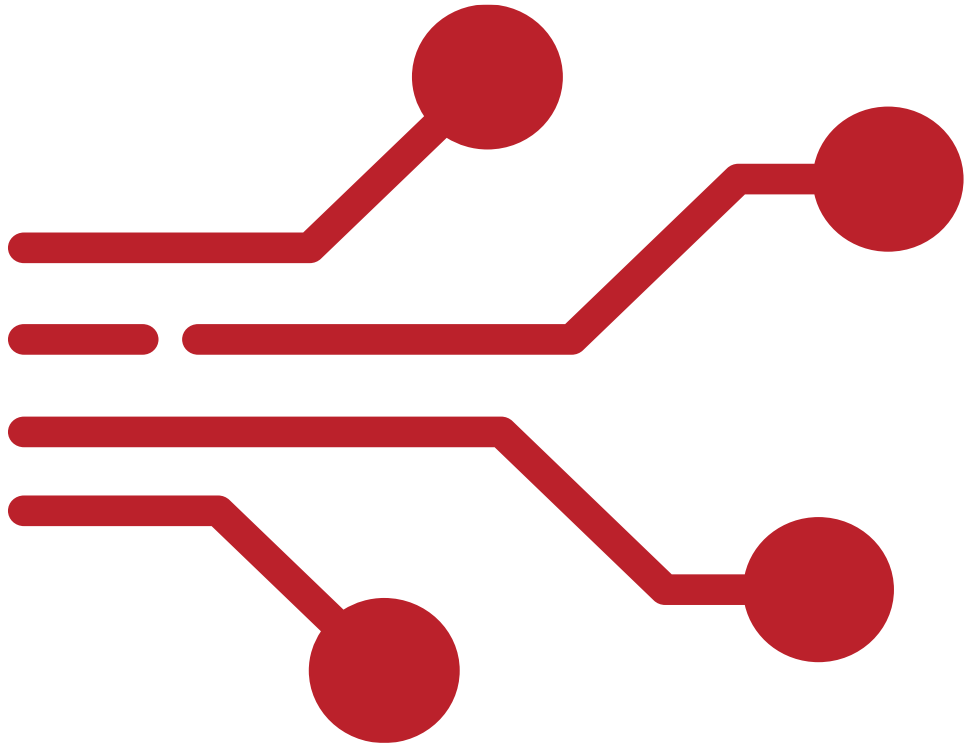
```
Console.Write("Podaj n: ");
int n = int.Parse(Console.ReadLine());
int suma4 = 0;

foreach (int liczba in Enumerable.Range(1, n))
{
    suma4 += liczba;
}
Console.WriteLine($"Suma (foreach): {suma4}");
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

119

Instrukcje sterujące



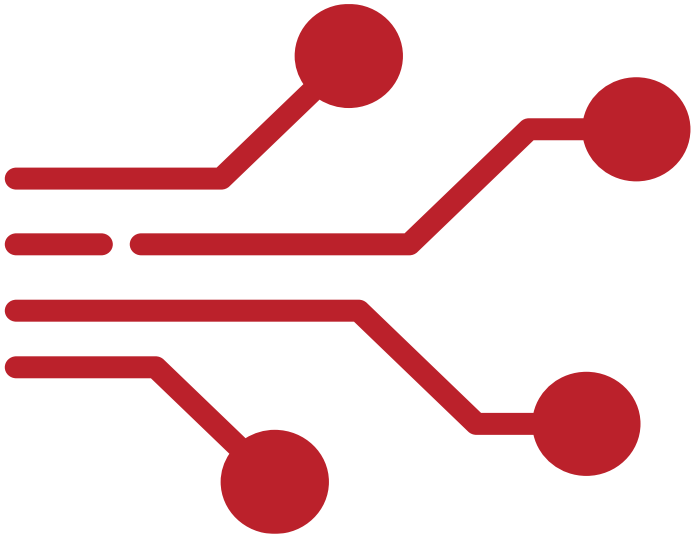
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

120

Instrukcja try...catch

try – blok kodu, w którym mogą wystąpić błędy.



catch – blok obsługi błędów (wyjątków).

Jeśli w try wystąpi wyjątek, to program przechodzi do catch.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

121

Przykład

```
try
{
    int x = int.Parse("abc"); // błąd
}
catch (FormatException e)
{
    Console.WriteLine("Niepoprawny format
    liczby!");
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

122

Przykład - ogólniej

```
try
{
    int x = int.Parse("abc"); // błąd
}
catch (Exception e)
{
    Console.WriteLine("Błąd!");
}
```

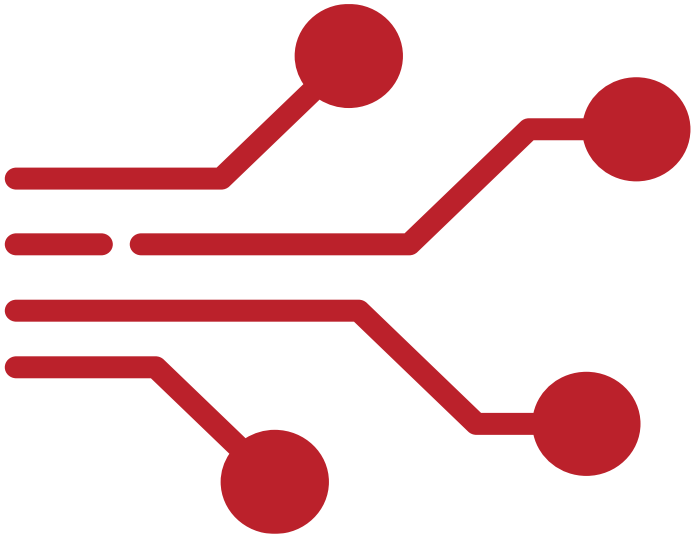

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

123

Blok finally

finally – blok, który wykona się zawsze, niezależnie od tego, czy był wyjątek, czy nie.



Jest stosowany do „sprzątania”: zamykanie plików, zwalnianie zasobów, przywracanie właściwego biegu sterowania.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

124

Przykład

```
try
{
    Console.WriteLine("Otwieram plik...");
    throw new Exception("Ups!");
}
catch (Exception)
{
    Console.WriteLine("Błąd przy pracy z plikiem.");
}
finally
{
    Console.WriteLine("Zamykam plik.");
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

125

Przykład

```
try
{
    Console.Write("Podaj liczbę: ");
    int liczba = int.Parse(Console.ReadLine());
    Console.WriteLine($"Liczba do kwadratu: {liczba * liczba}");
}
catch (FormatException)
{
    Console.WriteLine("To nie jest poprawna liczba!");
}
finally
{
    Console.WriteLine("Koniec programu.");
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

126

Przykład. Wariant

```
try
{
    Console.Write("Podaj liczbę: ");
    int liczba = int.Parse(Console.ReadLine());
    Console.WriteLine($"Liczba do kwadratu: {liczba * liczba}");
}
catch (FormatException e)
{
    Console.WriteLine("To nie jest poprawna liczba!");
    Console.WriteLine(e.Message);
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

127

Filtrowanie wyjątków

Jest to możliwość dodania warunku do bloku catch.

Dzięki temu wyjątek zostanie przechwycony tylko wtedy, gdy warunek jest spełniony.

Składnia:

```
catch (Exception e) when (warunek)
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

128

Przykład

```
try
{
    Console.Write("Podaj liczbę całkowitą: ");
    int n = int.Parse(Console.ReadLine());
    Console.WriteLine(100 / n);
}
catch (DivideByZeroException e) when (DateTime.Now.DayOfWeek ==
DayOfWeek.Monday)
{Console.WriteLine("W poniedziałki nie dzielimy przez zero!");}
catch (DivideByZeroException)
{Console.WriteLine("Dzielenie przez zero!");}
catch (FormatException)
{Console.WriteLine("To nie jest poprawna liczba!");}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

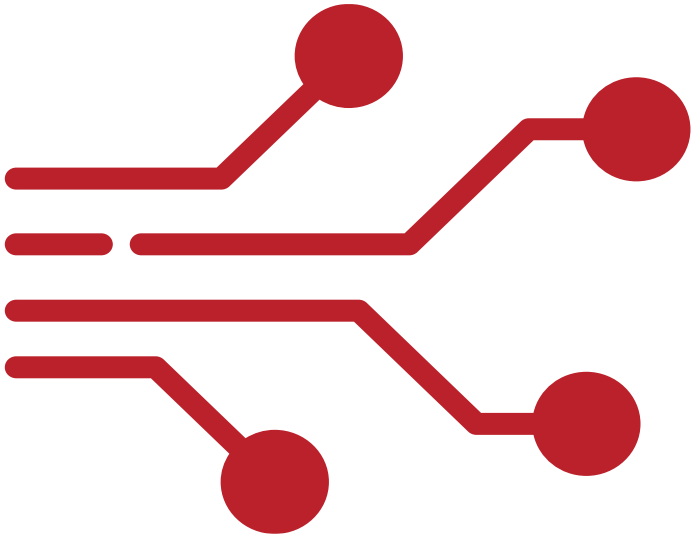
Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

129

Ważny wyjątek

NullReferenceException

Wyjątek rzucany w trakcie wykonywania programu, w przypadku wyłuskania (ang. *dereference*) wartości null, czyli przy próbach dostępu do zmiennej lub elementu, który ma wartość null.

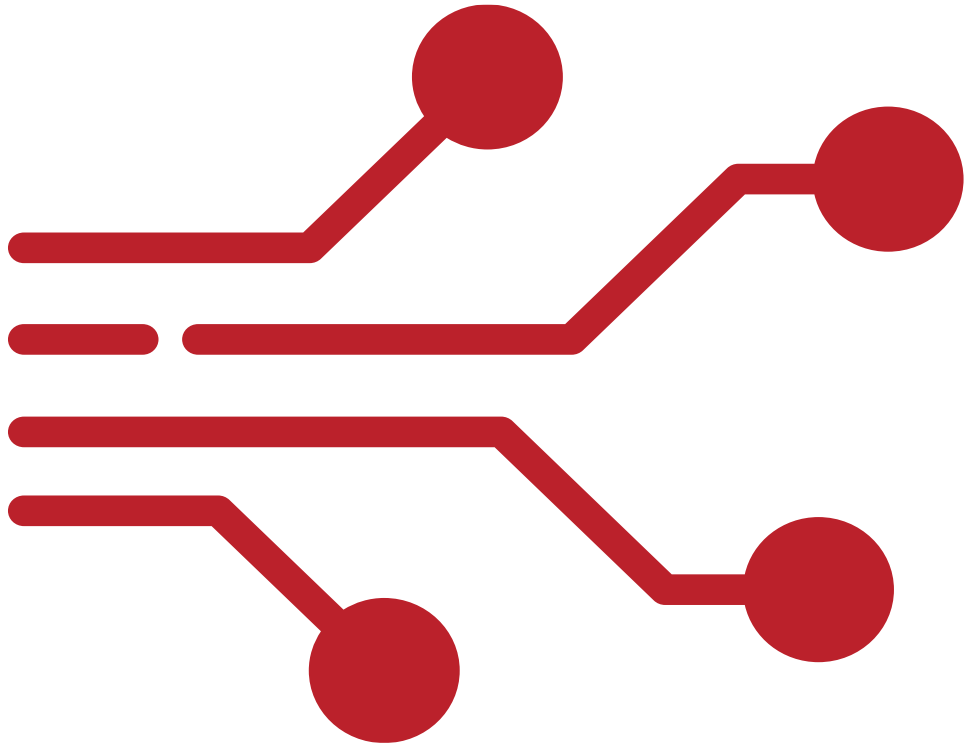


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

130

We/wy



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

131

Najważniejsze operacje

Console.Write() – wypisuje tekst bez przejścia do nowej linii.

Console.WriteLine() – wypisuje tekst i przechodzi do nowej linii.

```
Console.Write("Ala ");
Console.Write("ma kota ");
Console.WriteLine("i psa.");
// Wynik: Ala ma kota i psa.
int x = 5, y = 7;
Console.WriteLine($"Suma: {x + y}");
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

132

Podstawowe wejście

Console.ReadLine() – odczytuje cały wiersz wpisany przez użytkownika jako string.

Aby użyć jako liczby - potrzebne parsowanie (int.Parse, double.Parse, TryParse).

```
Console.Write("Podaj swoje imię: ");
string imie = Console.ReadLine();
Console.WriteLine($"Witaj, {imie}!");
Console.Write("Podaj wiek: ");
int wiek = int.Parse(Console.ReadLine());
Console.WriteLine($"Masz {wiek} lat.");
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

133

Obsługa błędów przy wejściu

Problem:

Użytkownik może wpisać niepoprawne dane (np. litery zamiast liczby), np. int.Parse("abc") wywoła wyjątek FormatException.

Rozwiązanie:

```
Console.Write("Podaj liczbę: ");
string tekst = Console.ReadLine();
if (int.TryParse(tekst, out int liczba))
    Console.WriteLine($"Kwadrat liczby: {liczba * liczba}");
else
    Console.WriteLine("To nie jest poprawna liczba!");
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

134

Zapis do pliku tekstowego

File.WriteAllText

Najprostsza metoda: zapisuje cały tekst do pliku.Jeśli plik istnieje, zostanie nadpisany.

```
string tekst = "Ala ma kota";
File.WriteAllText("dane.txt", tekst);
Console.WriteLine("Zapisano plik!");
```

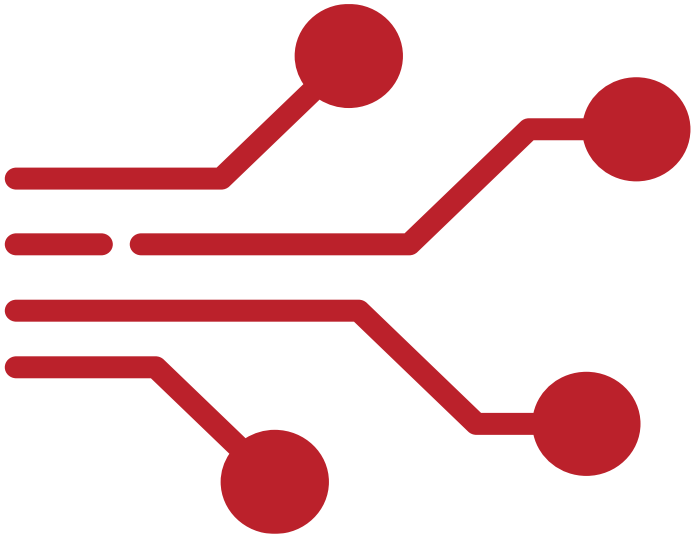
Dopisuje tekst na końcu pliku.

```
File.AppendAllText("dane.txt", "\nA kot ma Alę");
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

135



Odczyt z pliku tekstowego

File.ReadAllText - wczytuje całą zawartość pliku jako string.

```
string zawartosc = File.ReadAllText("dane.txt");
Console.WriteLine("Plik zawiera:");
Console.WriteLine(zawartosc);
```

File.ReadAllLines - zwraca tablicę linii (string[])

```
string[] linie = File.ReadAllLines("dane.txt");
foreach (string linia in linie)
Console.WriteLine(linia);
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

136

Przykład (plik, który w liniach ma 5, 12, 7, 3 – bez przecinków)

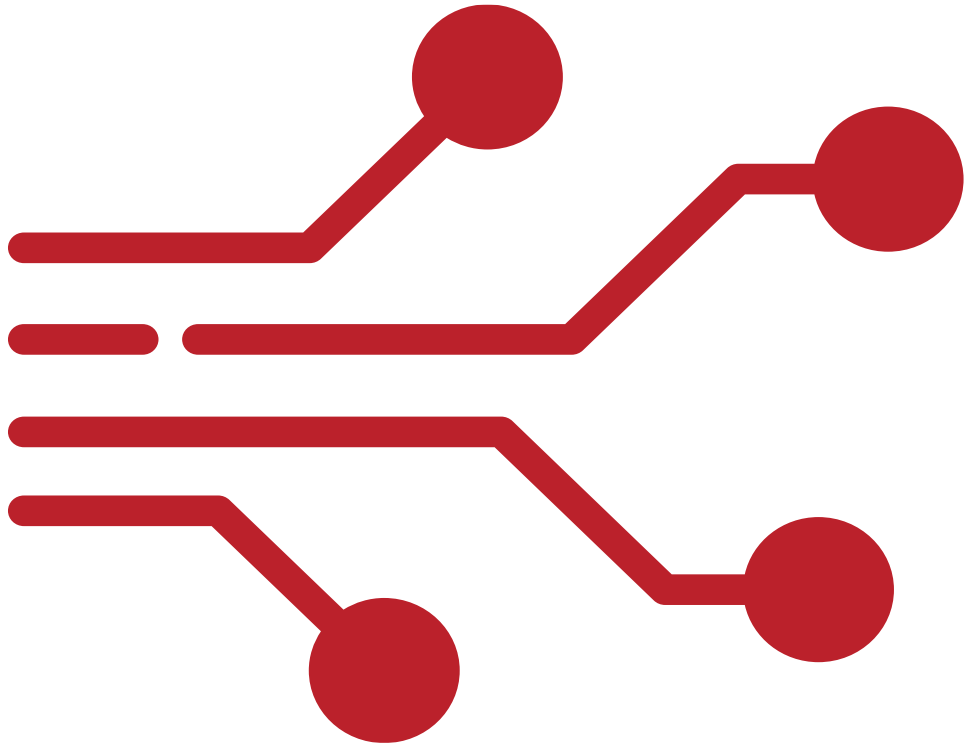
```
string[] linie =
File.ReadAllLines("liczby.txt");
int suma = 0;
foreach (string linia in linie)
{
    if (int.TryParse(linia, out int liczba))
        suma += liczba;
}
Console.WriteLine($"Suma liczb w pliku:{suma}");
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

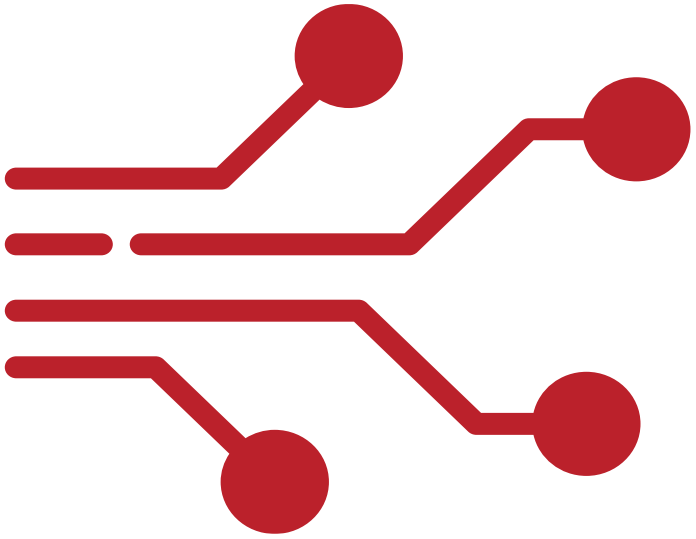
137

Debugowanie



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

138



Debugowanie

- Proces wyszukiwania i poprawiania błędów w kodzie.
- Umożliwia śledzenie działania programu krok po kroku.
- Pozwala sprawdzać wartości zmiennych w trakcie działania.

Dlaczego ważne?

- Błędy są naturalną częścią programowania.
- Debugger pozwala „zajrzeć do środka” programu i zrozumieć jego logikę.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00
Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

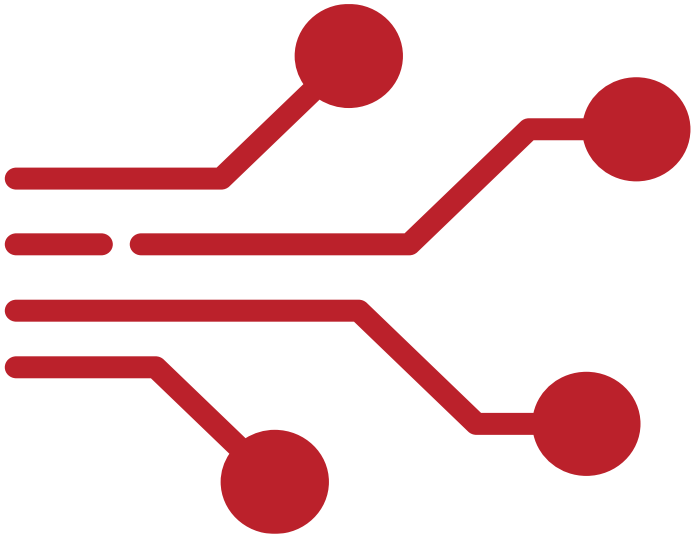
Punkty przerwania (breakpoints)

Jak działają?

- Zatrzymują program w wybranym miejscu.
- Od tego momentu możemy śledzić zmienne i wykonywać kod krokowo.

Jak ustawić?

- Kliknięcie w marginesie edytora (czerwona kropka).

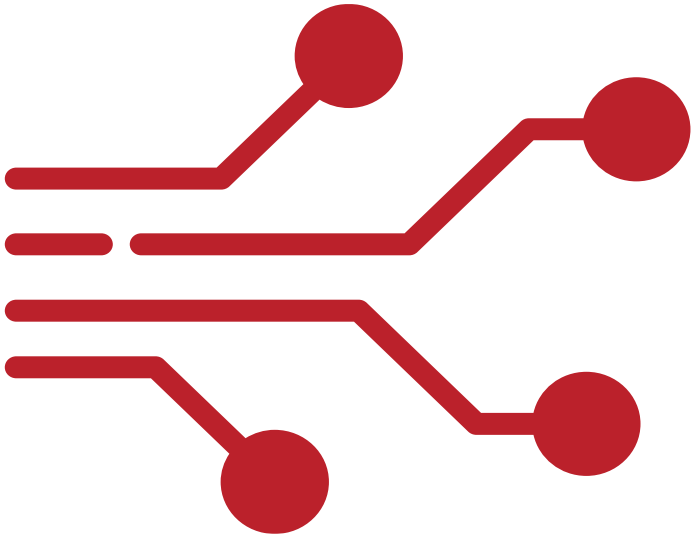


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

140

Okna debugera

- Lokalne (Locals) – pokazuje wszystkie zmienne w bieżącej metodzie.
- Watch – ręczne dodanie zmiennych do obserwacji.
- Call Stack – pokazuje stos wywołań metod (drogę programu).

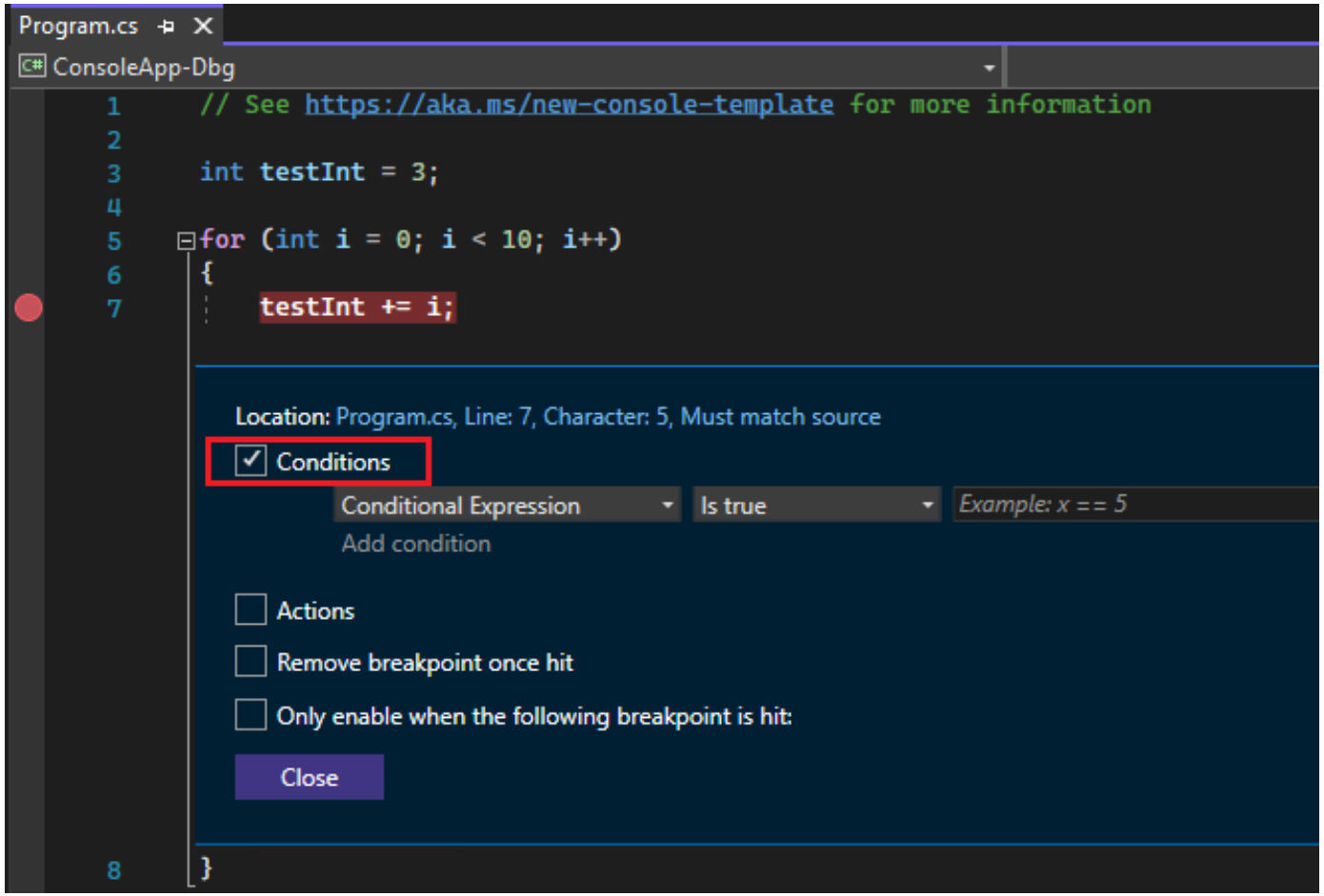


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

141

Punkty przerwania warunkowe



Rys. 12 Punkty przerwania warunkowe
[źródło: Dokumentacja Visual Studio]
https://learn.microsoft.com/en-us/visualstudio/debugger/using-breakpoints?view=vs-2022&pivots=programming-language-dotnet

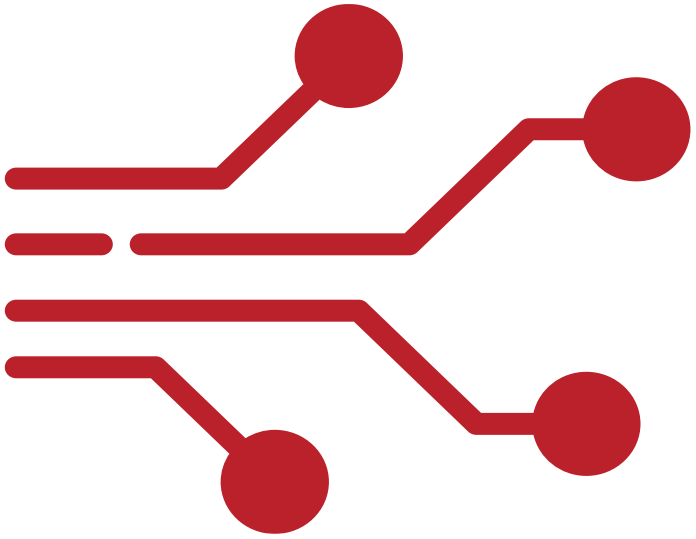
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Hot reload

Funkcja w Visual Studio i .NET, która pozwala:

- zmieniać kod w trakcie działania programu,
- zobaczyć efekty bez restartowania aplikacji.

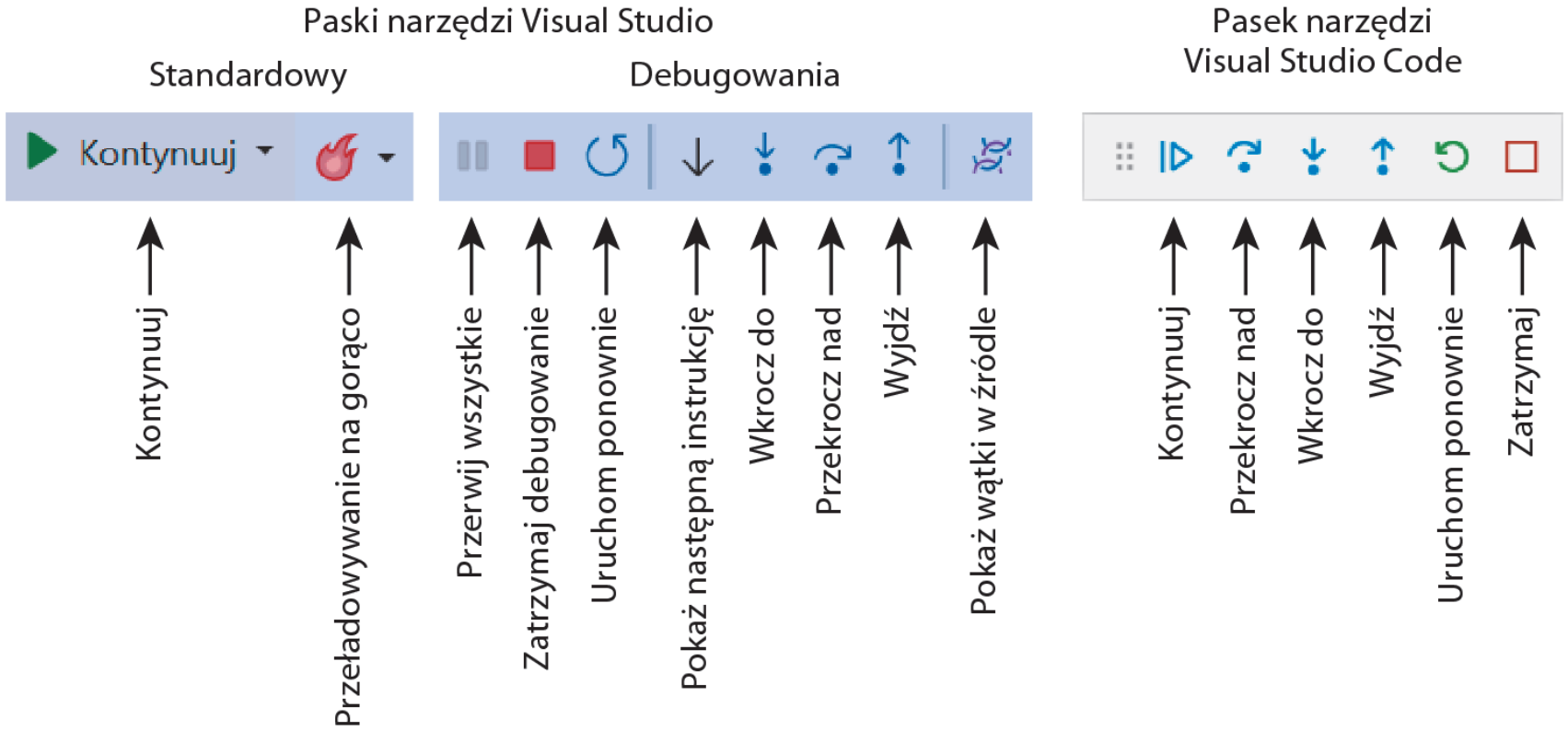


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

143

Pasek narzędzi



Rys. 13 Paski narzędzi debuggera
[źródło: M. J. Price, C# 12 i .NET 8 dla programistów aplikacji wieloplatformowych, Helion 2024]

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

144

Dziękuję za uwagę

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Wykład nr 4
Temat: Funkcje i modularność programu. Definiowanie i wywoływanie funkcji. Parametry i wartości zwracane

Paweł Karczmarek
Katedra Inteligencji Obliczeniowej Politechniki Lubelskiej

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Plan prezentacji

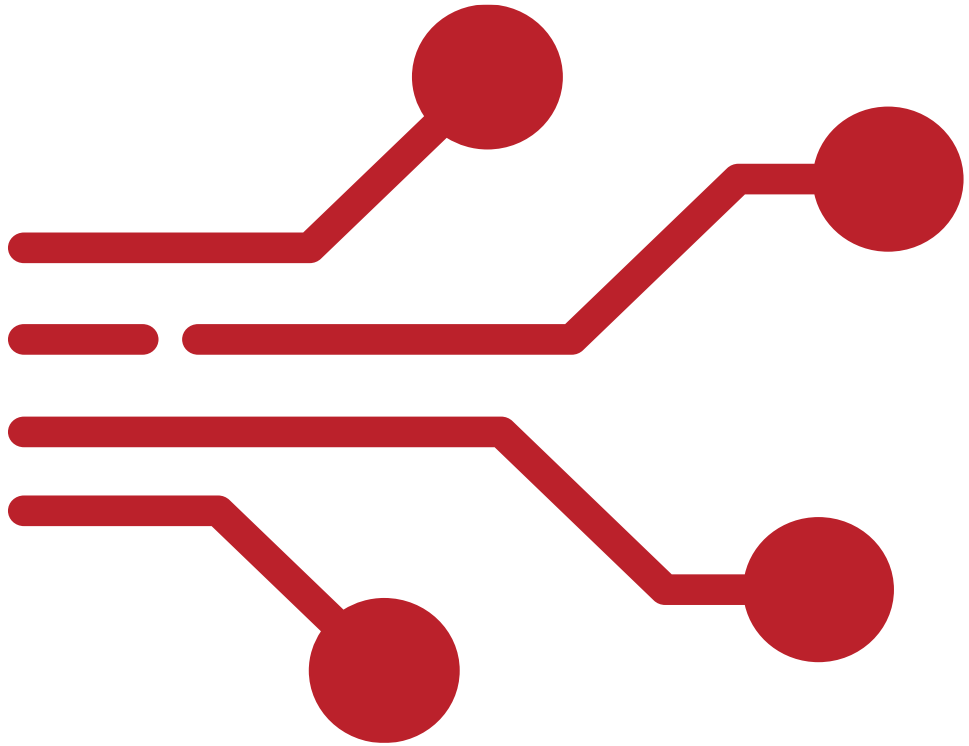
- Modularność i funkcje
- Definiowanie funkcji
- Parametry funkcji
- Wartości zwracane
- Zaawansowane aspekty

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

147

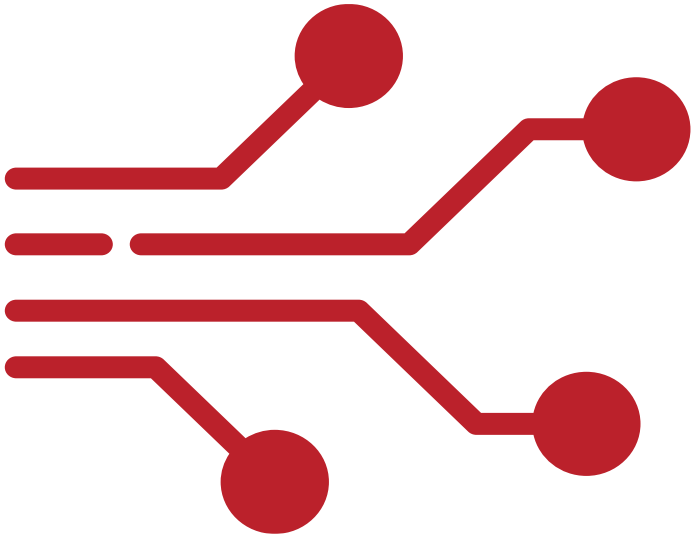
Modularność i funkcje



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

148



Czym jest funkcja?

Funkcja (metoda) to wydzielony fragment kodu, który można wielokrotnie wywołać w programie. Przyjmuje dane wejściowe (parametry) i może zwracać wynik (wartość). Umożliwia modularność – dzielenie programu na mniejsze, logiczne części.

Cechy funkcji:

- Ma nazwę, która identyfikuje jej działanie.
- Określa typ zwracanej wartości (lub void, jeśli nic nie zwraca).
- Może mieć zero, jeden lub wiele parametrów.
- Jej kod wykonuje się dopiero po wywołaniu.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

149

Przykład

```
int Dodaj(int a, int b)
{
    return a + b;
}
```

- Dodaj – nazwa funkcji
- int – typ zwracanej wartości
- (int a, int b) – lista parametrów
- return a + b; – instrukcja zwracająca wynik

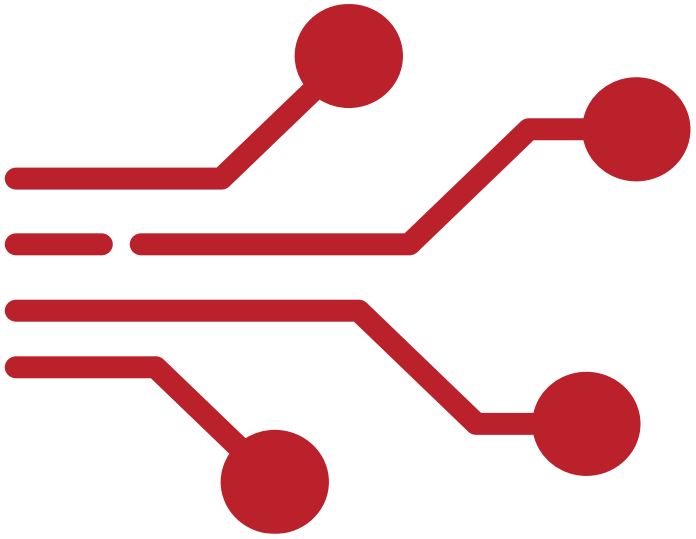
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

150

Modularność programu

- Podział programu na mniejsze części (funkcje, klasy, moduły).
- Każdy moduł odpowiada za jedno wyraźne zadanie (Single Responsibility Principle).
- Moduły można rozwijać i testować niezależnie.



Zalety modularności:

- Reużywalność – jedna funkcja może być używana w wielu miejscach.
- Łatwiejsze testowanie i debugowanie – można sprawdzać małe fragmenty zamiast całego programu.
- Czytelność i przejrzystość – łatwiej zrozumieć strukturę kodu.
- Szybszy rozwój i utrzymanie – łatwiej zmienić pojedynczy moduł niż cały system.
- Praca zespołowa – różne osoby mogą rozwijać różne moduły równolegle.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

151

Programy najwyższego poziomu i funkcje lokalne

Program najwyższego poziomu (top-level statements):

- Od C# 9 można pisać program bez ręcznego tworzenia klasy Program i metody Main().
- Kompilator sam generuje brakujące elementy.
- Ułatwia szybkie pisanie krótkich programów i prototypów.

Funkcje lokalne:

- Funkcje zagnieżdżone wewnątrz innej funkcji.
- Mają dostęp do zmiennych lokalnych funkcji nadrzędnej.
- Poprawiają czytelność kodu – użyte tylko w jednym miejscu, nie „zaśmiecają” całej klasy.

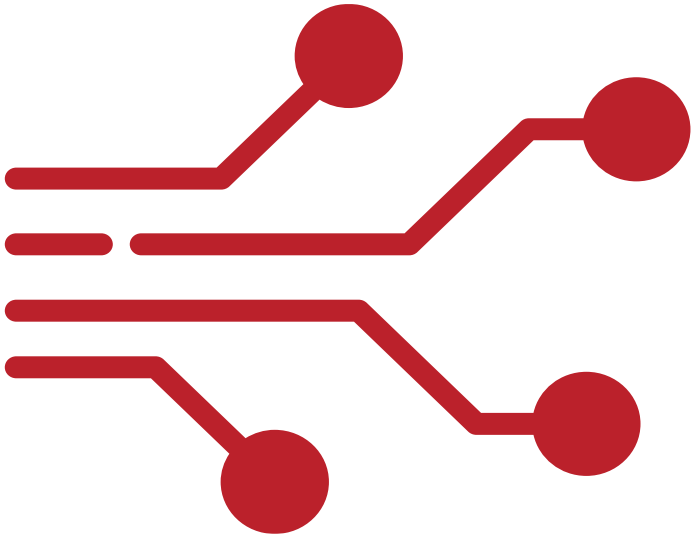
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

152

Przykład

```
// Program najwyższego poziomu
Console.WriteLine(CzyParzysta(10));
// wypisze True


bool CzyParzysta(int n)
// funkcja lokalna
{
    return n % 2 == 0;
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

153

Debugowanie funkcji w Visual Studio – szczegóły

Podstawowe narzędzia debuggera:

- Punkt przerwania (Breakpoint) – kliknięcie w lewym marginesie edytora (obok numeru linii) zatrzymuje program w wybranym miejscu.
- Step Into (F11) – pozwala wejść do wnętrza funkcji i śledzić jej wykonanie linijka po linijce.
- Step Over (F10) – wykonuje całą funkcję „w tle”, bez wchodzenia do środka.
- Step Out (Shift+F11) – kończy aktualną funkcję i wraca do miejsca, skąd została wywołana.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

154

Debugowanie funkcji w Visual Studio – szczegóły

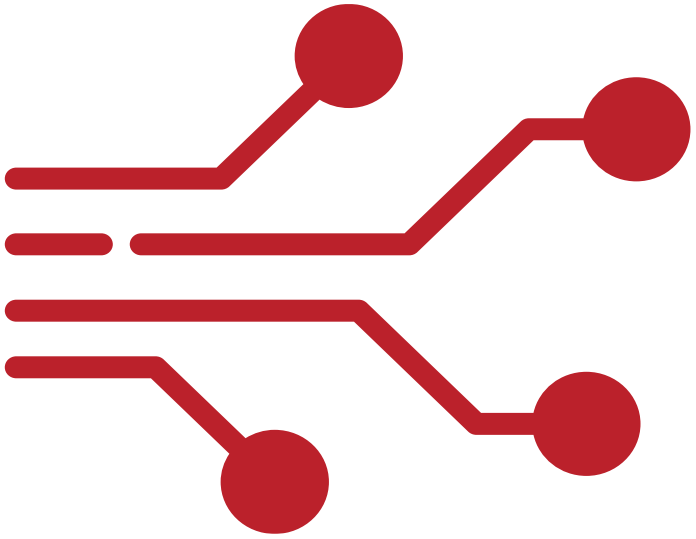
Okna pomocnicze w trakcie debugowania:

- Locals – pokazuje wszystkie lokalne zmienne aktualnie wykonywanej funkcji.
- Watch – pozwala dodać własne wyrażenia do obserwowania (np. a + b w funkcji Dodaj).
- Call Stack – wyświetla stos wywołań, czyli ścieżkę funkcji prowadzącą do aktualnego miejsca.
- Immediate Window – można tam wpisywać kod C# i od razu zobaczyć wyniki (np. wywołać Dodaj(2,3) podczas przerwania programu).

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Debugowanie funkcji w Visual Studio – szczegóły



Dodatkowe możliwości:

- Conditional Breakpoint – punkt przerwania, który działa tylko, jeśli spełniony jest warunek (np. zatrzymaj, jeśli x == 0).
- Data Breakpoint – program zatrzyma się, gdy zmienna przyjmie nową wartość.
- Hover Inspect – najechanie kursorem na zmienną pokazuje jej bieżącą wartość w dymku.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

156

Składnia funkcji

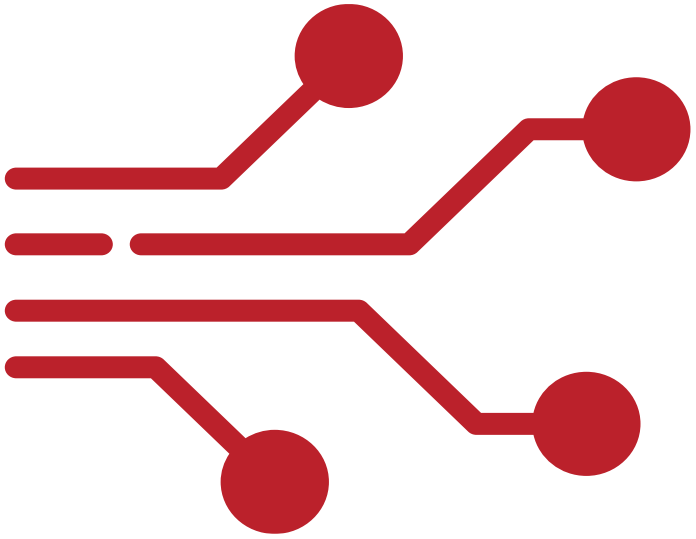
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

157

Składnia funkcji w C#

Podstawowa budowa funkcji (metody):

```
[atrybuty] [modyfikatory] typ_zwracany
NazwaFunkcji(lista_parametrów)
{
    // ciało funkcji – instrukcje, które zostaną wykonane
    return wartość; // jeśli funkcja coś zwraca
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

158

Składnia funkcji w C#

Elementy składni:

- Modyfikatory dostępu – public, private, internal, protected
- Modyfikatory dodatkowe – static, virtual, override, async
- Typ zwracany – np. int, bool, string, void (gdy nic nie zwracamy)
- Nazwa funkcji – zgodna z konwencją PascalCase (ew. camelCase)
- Lista parametrów – w nawiasach (), może być pusta lub zawierać wiele parametrów
- Ciało funkcji – blok { ... } zawierający instrukcje

```
public static int Dodaj(int a, int b)
{
    return a + b;
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

159

Funkcje statyczne i lokalne

Funkcje statyczne:

- Deklarowane ze słowem kluczowym static.
- Należą do klasy, a nie do konkretnego obiektu.
- Wywołujemy je poprzez nazwę klasy (np. Math.Sqrt(9)).
- Typowe dla narzędziowych operacji, które nie potrzebują stanu obiektu.

Funkcje lokalne:

- Mogą być zagnieżdżone w innej funkcji (od C# 7).
- Mają dostęp do zmiennych lokalnych funkcji nadrzędnej.
- Poprawiają czytelność kodu, jeśli dana funkcja jest pomocnicza i potrzebna tylko w jednym miejscu.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

160

Przykład

```
class Kalkulator
{
    public static int Dodaj(int a, int b) // funkcja statyczna
    {
        return a + b;
    }
}
Console.WriteLine(Kalkulator.Dodaj(3, 4)); // użycie
// Funkcja lokalna
int Kwadrat(int x)
{
    int Pomnoz(int y) => y * y; // lokalna
    return Pomnoz(x);
}
Console.WriteLine(Kwadrat(5)); // 25
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

161

Funkcje zwracające wartość

- Funkcje, które po zakończeniu działania oddają wynik za pomocą instrukcji return.
- Typ zwracany musi być zgodny z deklaracją w nagłówku funkcji.
- Wartość zwrócona może zostać:
  - przypisana do zmiennej,
  - użyta w wyrażeniu,
  - przekazana jako argument innej funkcji.

Zasady:

- Funkcja musi mieć zadeklarowany typ zwracany inny niż void.
- Musi istnieć przynajmniej jedna ścieżka, która kończy się instrukcją return.

* Wartość zwracana staje się wynikiem wywołania funkcji.

```
public static double ObliczSrednia(int a, int b){return (a + b) / 2.0;}
double wynik = ObliczSrednia(4, 6); // wynik = 5.0
Console.WriteLine($"Średnia to: {wynik}");
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

162

Funkcje void vs funkcje zwracające wartość

Funkcje void:

- Deklarowane ze słowem void.
- Nie zwracają wartości – wykonują tylko akcję (np. wyświetlają tekst, zapisują do pliku).
- Nie mogą używać instrukcji return z wartością (mogą użyć samego return jako „wyjścia”).

Funkcje zwracające wartość:

- Deklarują typ zwracany (np. int, string, double).
- Muszą zakończyć się instrukcją return zwracającą zgodny typ.
- Wynik można przypisać do zmiennej lub użyć w innym wyrażeniu.

```
// Funkcja void – tylko efekt uboczny

public static void Przywitaj(string imie){Console.WriteLine($"Cześć, {imie}!");}

// Funkcja zwracająca wartość

public static int Pomnoz(int a, int b){return a * b;}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

163

Dokumentowanie funkcji komentarzami XML

Czym są komentarze XML?

- Specjalne komentarze, które generują dokumentację funkcji.
- Mogą zawierać znaczniki XML (<summary>, <param>, <returns>).
- Wspierane przez Visual Studio – wyświetlają się jako podpowiedzi IntelliSense.
- Mogą być automatycznie eksportowane do plików pomocy (np. .xml lub .chm).

Najczęściej używane znaczniki XML:

- <summary> – krótki opis funkcji.
- <param name="..."> – opisuje parametr.
- <returns> – opis zwracanej wartości.
- <remarks> – dodatkowe uwagi.
- <example> – przykładowe użycie funkcji.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

164

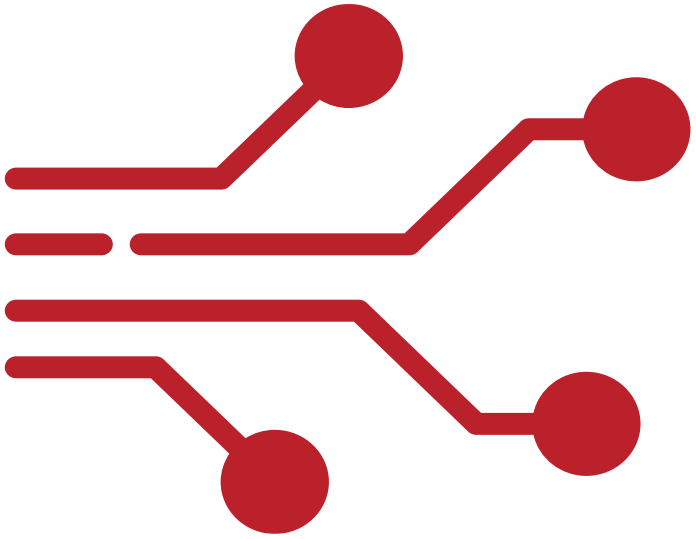
Przykład

```
/// <summary>
/// Oblicza średnią dwóch liczb całkowitych.
/// </summary>
/// <param name="a">Pierwsza liczba</param>
/// <param name="b">Druga liczba</param>
/// <returns>Średnia arytmetyczna liczb a i b</returns>
public static double ObliczSrednia(int a, int b)
{
    return (a + b) / 2.0;
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

165



Wyrażenia lambda jako forma funkcji

Wyrażenie lambda:

- Krótka, zwięzła forma zapisu funkcji anonimowej (bez nazwy).
- Używa operatora => („idzie do”).
- Może być przypisana do zmiennej lub przekazana jako parametr innej funkcji.
- Szeroko stosowana w LINQ i programowaniu funkcyjnym w C#.

Składnia:

(parametry) => wyrażenie

(parametry) => { instrukcje }

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Przykłady

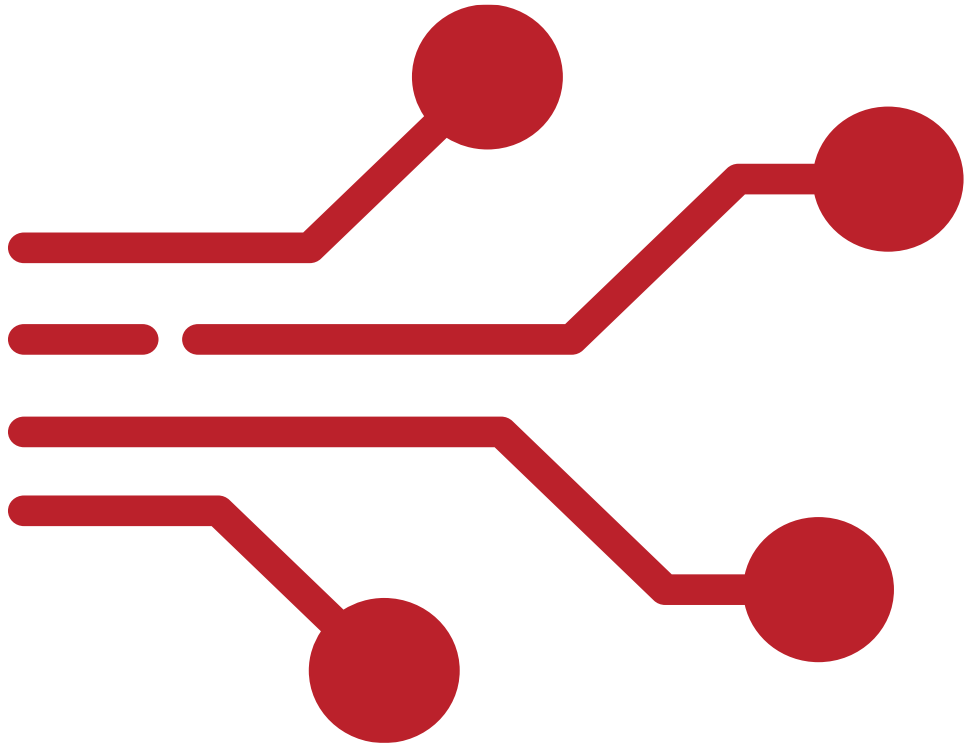
```
Func<int, int, int> Dodaj = (a, b) => a + b;
Console.WriteLine(Dodaj(3, 4)); // 7
Func<int, int> KwadratLubZero = x =>
{
    if (x < 0) return 0;
    return x * x;
};
Console.WriteLine(KwadratLubZero(5));  // 25
Console.WriteLine(KwadratLubZero(-3)); // 0
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

167

Parametry funkcji



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

168

Parametry i argumenty – terminologia

Parametry:

- Zmiennie deklarowane w nagłówku funkcji.
- Określają, jakie dane funkcja może przyjąć.
- Każdy parametr ma swój typ i nazwę.
- Są jak miejsce w pudełku na dane wejściowe.

Argumenty:

- Konkretne wartości, które przekazujemy podczas wywołania funkcji.
- Mogą być liczbami, tekstami, zmiennymi, obiektami itp.
- Są dopasowywane do parametrów funkcji.

```
int Pomnoz(int a, int b) // a, b parametry
{return a * b;}
int wynik = Pomnoz(3, 4); // 3 i 4 argumenty
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

169

Przekazywanie przez wartość

- Domyślny sposób przekazywania danych w C#.
- Do funkcji trafia kopiowana wartość argumentu, a nie oryginalna zmienna.
- Zmiany wykonane na parametrze nie wpływają na wartość zmiennej poza funkcją.

```
void Zmien(int x){
    x = x + 10;
    Console.WriteLine($"Wewnątrz funkcji: {x}");}
int liczba = 5;
Zmien(liczba);
Console.WriteLine($"Po wywołaniu: {liczba}");
```

Wynik:

Wewnątrz funkcji: 15

Po wywołaniu: 5

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Przekazywanie przez referencję (ref, in, out)

- Funkcja pracuje nie na kopii, ale na oryginalnej zmiennej.
- Parametr i argument wskazują na to samo miejsce w pamięci.
- Zmiana wewnątrz funkcji jest widoczna poza nią.

Słowa kluczowe:

- ref – przekazanie przez referencję (zmienna musi być wcześniej zainicjalizowana).
- out – przekazanie przez referencję, gdy funkcja ma zwrócić wartość w parametrze (zmienna nie musi być zainicjalizowana).
- in – przekazanie przez referencję tylko do odczytu (chroni przed zmianami).

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

171

Przykład

```
// ref
void Zwieksz(ref int x) { x++; }
int a = 5;
Zwieksz(ref a);
Console.WriteLine(a); // 6
// out
bool ProbaParsowania(string txt, out int wynik){
    return int.TryParse(txt, out wynik);}
if (ProbaParsowania("123", out int liczba))
    Console.WriteLine(liczba); // 123
// in
void Pokaz(in int y){
    Console.WriteLine(y); // można tylko odczytać}
int b = 10;
Pokaz(in b);
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

172

Parametry opcjonalne

- Parametry, dla których podajemy wartość domyślną w nagłówku funkcji.
- Jeśli przy wywołaniu funkcji nie podamy argumentu, to używana jest ta wartość.
- Ułatwiają pisanie bardziej elastycznych funkcji bez przeciążania metod.

Składnia:

```
typ Nazwa(parametr1, typ2 parametr2 = wartość_domyslna)
```

```
void Przywitaj(string imie, string powitanie = "Cześć")
{
    Console.WriteLine($"{powitanie}, {imie}!");
}
Przywitaj("Kuba");          // użyje wartości domyślnej -> "Cześć, Kuba!"
Przywitaj("Ania", "Hejka"); // użyje podanego argumentu -> "Hejka, Ania!"
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00
Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

173

Parametry nazwane

- Argumenty można przekazywać nie tylko według kolejności, ale także po nazwie parametru.
- Zwiększają czytelność wywołań funkcji.
- Pozwalają pominąć niektóre parametry opcjonalne.

Składnia:

```
NazwaParametru: wartość
```

```
void Zarejestruj(string imie, string nazwisko, int wiek = 18)
{Console.WriteLine($"{imie} {nazwisko}, wiek: {wiek}");}
// Wywołania:
Zarejestruj("Jan", "Kowalski");
Zarejestruj(nazwisko: "Nowak", imie: "Anna");
Zarejestruj(imie: "Ola", nazwisko: "Malinowska", wiek: 25);
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

174

Mieszanie parametrów

Parametry nazwane:

- Można podać argumenty po nazwie zamiast w kolejności.
- Ułatwia czytelność, zwłaszcza gdy parametry są tego samego typu.

Mieszanie sposobów:

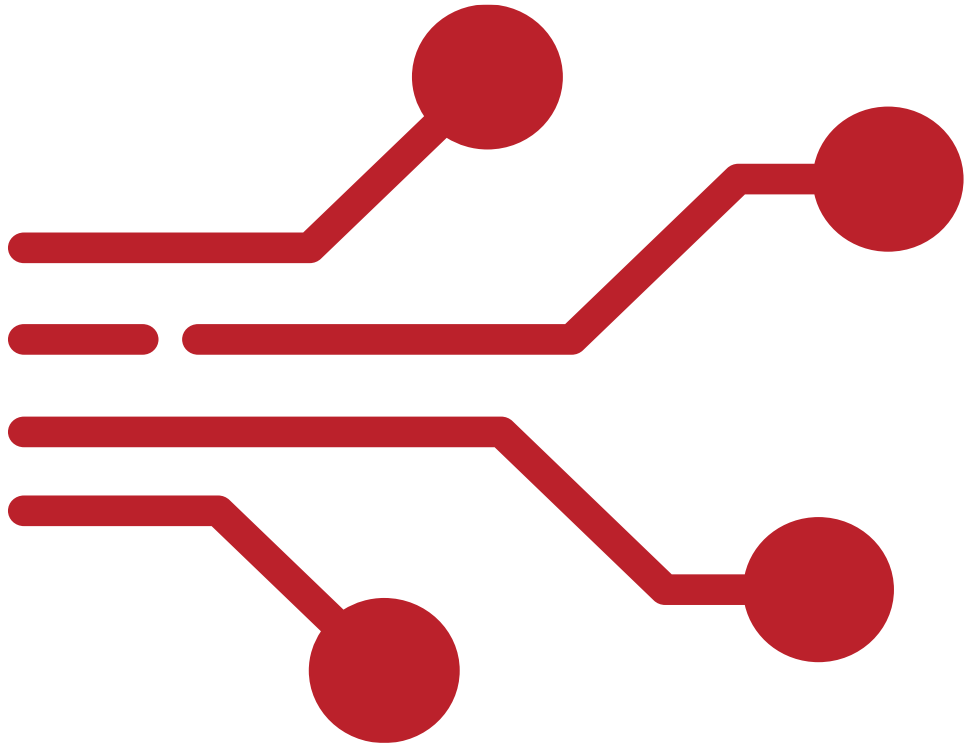
- Możemy łączyć argumenty pozycyjne i nazwane.
- Zasada: najpierw parametry pozycyjne, potem nazwane.
- Dzięki temu można pominąć niektóre parametry opcjonalne.
- Parametry opcjonalne muszą być zadeklarowane na końcu listy parametrów.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

175

Wartości zwracane



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

176

Instrukcja return i typ void

Instrukcja return:

- Kończy działanie funkcji.
- Zwraca wartość (jeśli funkcja ma typ inny niż void).
- W funkcjach void można użyć return; jako „wyjścia awaryjnego”, ale bez wartości.
- W funkcjach z typem zwracanym każda ścieżka musi kończyć się return.

Typ void:

- Oznacza, że funkcja nie zwraca żadnej wartości.
- Funkcja void wykonuje tylko akcję (np. wypisuje coś, zapisuje do pliku).
- return jest opcjonalny – może nie występować w ogóle.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

177

Przykłady

```
// Funkcja zwracająca wartość
int Dodaj(int a, int b)
{
    return a + b; // MUSI zwrócić int
}
// Funkcja void
void PowiedzCzesc(string imie)
{
    if (string.IsNullOrEmpty(imie))
        return; // wyjście awaryjne
    Console.WriteLine($"Cześć, {imie}!");
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

178

Zasięg i czas życia zmiennych w funkcjach

Zasięg zmiennych:

- Określa, gdzie w kodzie dana zmienna jest widoczna i dostępna.
- Zmienna zadeklarowana wewnątrz funkcji jest widoczna tylko w tej funkcji.
- Zmienna zadeklarowana w bloku { ... } jest widoczna tylko w tym bloku.

Czas życia zmiennych:

- Zmienna lokalna istnieje tylko podczas działania funkcji.
- Po zakończeniu funkcji pamięć jest zwalniana.
- Zmienna statyczna (static) ma dłuższy czas życia — istnieje przez cały czas działania programu.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

179

Przykład

```
void Test()
{
    int a = 5; // zmienna lokalna funkcji
    if (a > 0)
    {
        int b = 10; // zmienna lokalna bloku if
        Console.WriteLine(a + b);
    }
    // Console.WriteLine(b); // b już nie istnieje
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

180

Zwracanie wartości przez ref

- Funkcja może zwrócić referencję do zmiennej, a nie jej kopię.
- Pozwala modyfikować oryginalną zmienną poza funkcją.
- Wartość zwrócona musi być przypisana do ref lub użyta odczytowo.

Składnia:

```
ref typ NazwaFunkcji(...) { ... }
ref var zmienna = ref Funkcja();
```

Przykład:

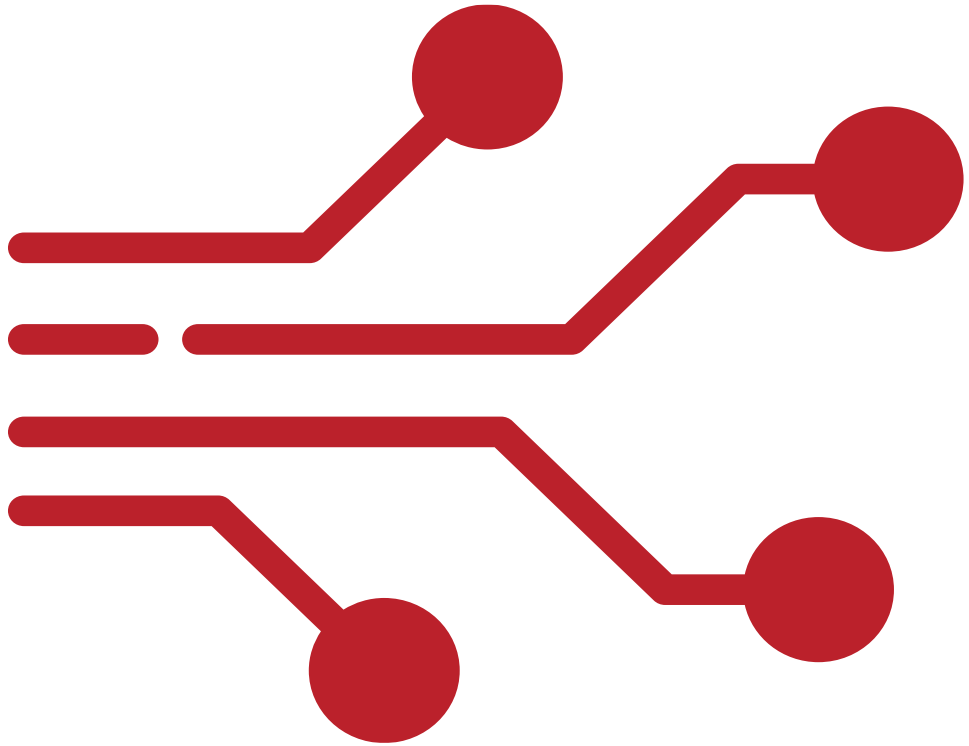
```
ref int ZnajdzElement(int[] tablica, int indeks){
    return ref tablica[indeks];}
int[] dane = { 1, 2, 3 };
ref int elem = ref ZnajdzElement(dane, 1);
elem = 42;  // zmienia oryginalną tablicę
Console.WriteLine(dane[1]); // 42
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

181

Aspekty zaawansowane



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

182

Przeciążanie metod (Overloading)

Na czym polega?

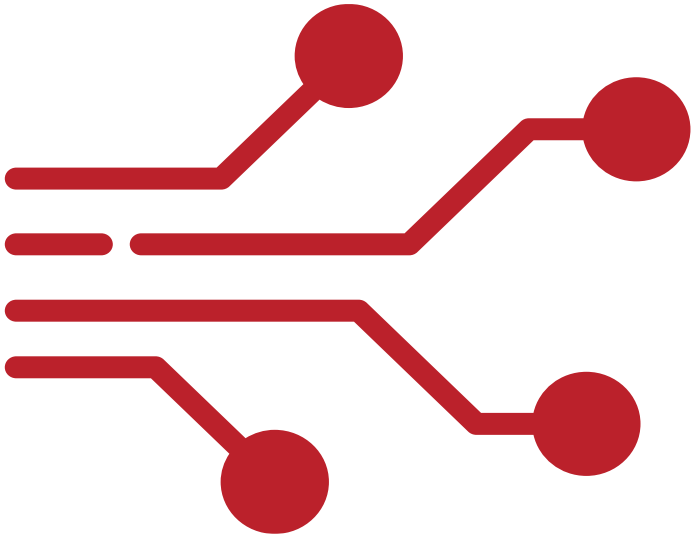
- Można definiować kilka metod o tej samej nazwie, ale z różnymi parametrami.
- Różnice: liczba parametrów, typy parametrów, kolejność parametrów.
- Nie można przeciążać tylko po typie zwracanym.

Przykład:

```
int Dodaj(int a, int b) => a + b;
double Dodaj(double a, double b) => a + b;
string Dodaj(string a, string b) => a + b;
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00
Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

183



Metody ogólne (generyczne)

- Funkcje, które działają z różnymi typami, bez duplikowania kodu.
- Używają parametrów typów <T>.
- Mogą mieć ograniczenia (where T : ...).

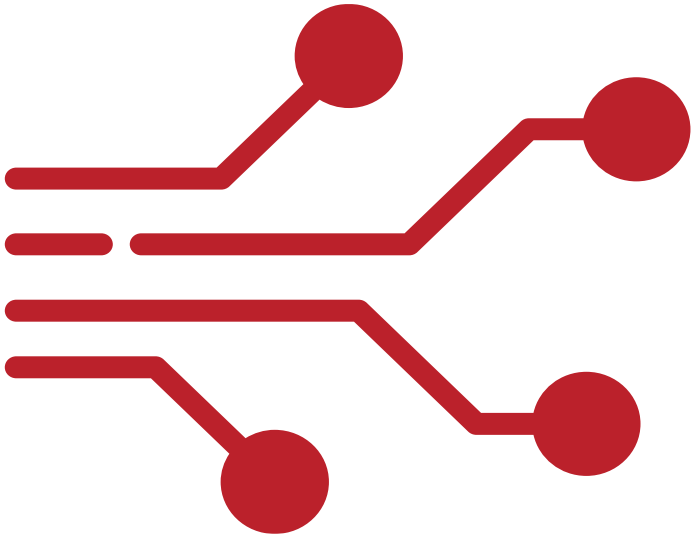
Przykład:

```
public static T Maksimum<T>(T a, T b) where T : IComparable<T>
{
    return a.CompareTo(b) > 0 ? a : b;
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

184



Delegaty i Func/Action

Delegaty:

- Typy reprezentujące odwołania do metod.
- Pozwalają traktować funkcje jak dane i przekazywać je jako argumenty.

Func i Action:

- Func<T1,...,TResult> – delegat zwracający wartość.
- Action<T1,...> – delegat bez zwracanej wartości (void).

```
Func<int, int, int> dodaj = (a, b) => a + b;
Action<string> wypisz = x => Console.WriteLine(x);
Console.WriteLine(dodaj(2, 3)); // 5
wypisz("Hello");
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

185

Wyrażenia lambda – składnia i idea

- Krótki, nowoczesny zapis funkcji anonimowej (czyli funkcji bez nazwy).
- Wykorzystuje operator => (tzw. „goes to”).
- Stosowana tam, gdzie potrzebujemy logiki w locie, bez pisania pełnej metody.

Postacie:

- Jednowyrażeniowa:

```
(x, y) => x + y
```

- Wieloliniowa:

```
(x, y) =>
{
    int wynik = x * y;
    return wynik;
}
```

Jeden parametr (nawias opcjonalny):

```
x => x * x
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

186

SRP (Single Responsibility Principle):

- Każda jednostka kodu powinna mieć jedną odpowiedzialność.
- Funkcje powinny być małe i skupione na jednym zadaniu.
- Ułatwia testowanie, debugowanie, ponowne użycie.

Przykład (złe podejście):

```
void PrzetworzDane()
{
    WczytajZPliku();
    PoliczStatystyki();
    ZapiszDoBazy();
}
```

Przykład (dobre podejście):

```
void WczytajZPliku() { ... }
void PoliczStatystyki() { ... }
void ZapiszDoBazy() { ... }
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

187

Dziękuję za uwagę

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Wykład nr 5
Temat: Rekurencja

Paweł Karczmarek
Katedra Inteligencji Obliczeniowej Politechniki Lubelskiej

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Plan prezentacji

- Koncepcja
- Rekurencja w praktyce

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

190

Koncepcja

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

191

Czym jest rekurencja?

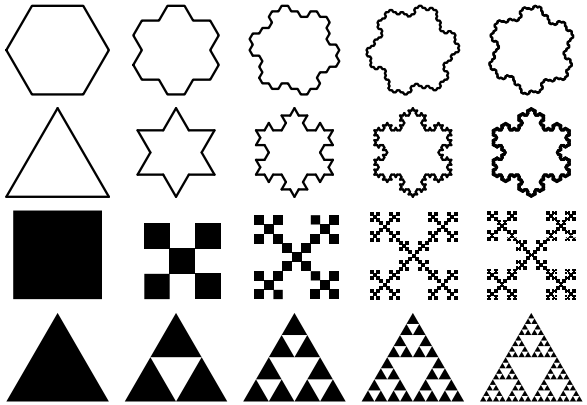
- To funkcja wywołująca samą siebie.
- Dwa filary: warunek bazowy (stop) + krok rekurencyjny (redukcja).
- Korzyści wynikające ze stosowania: kod bliski definicjom (matematyka, drzewa, grafy).
- Ryzyko podczas stosowania: przepełnienie stosu, koszty wydajności.

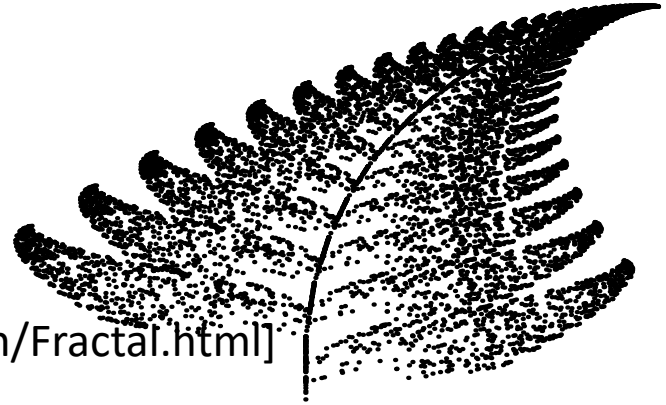
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

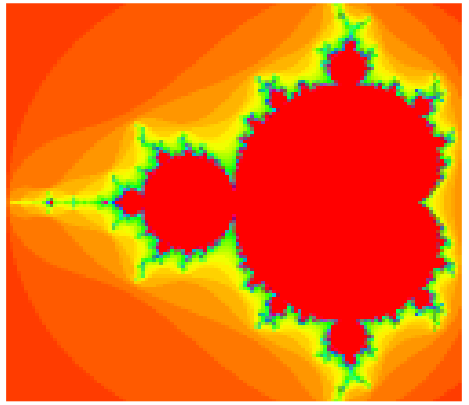
Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

192

Fraktale







Rys. 14 Fraktale
[źródło: https://mathworld.wolfram.com/Fractal.html]

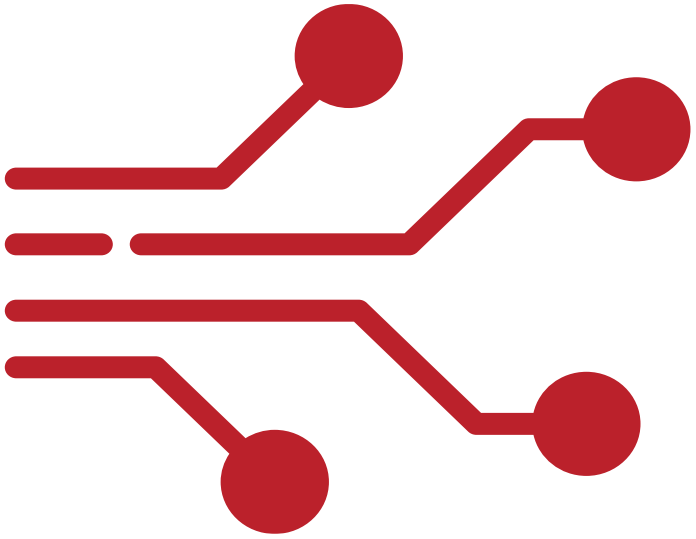
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

193

Szkielet funkcji rekurencyjnej

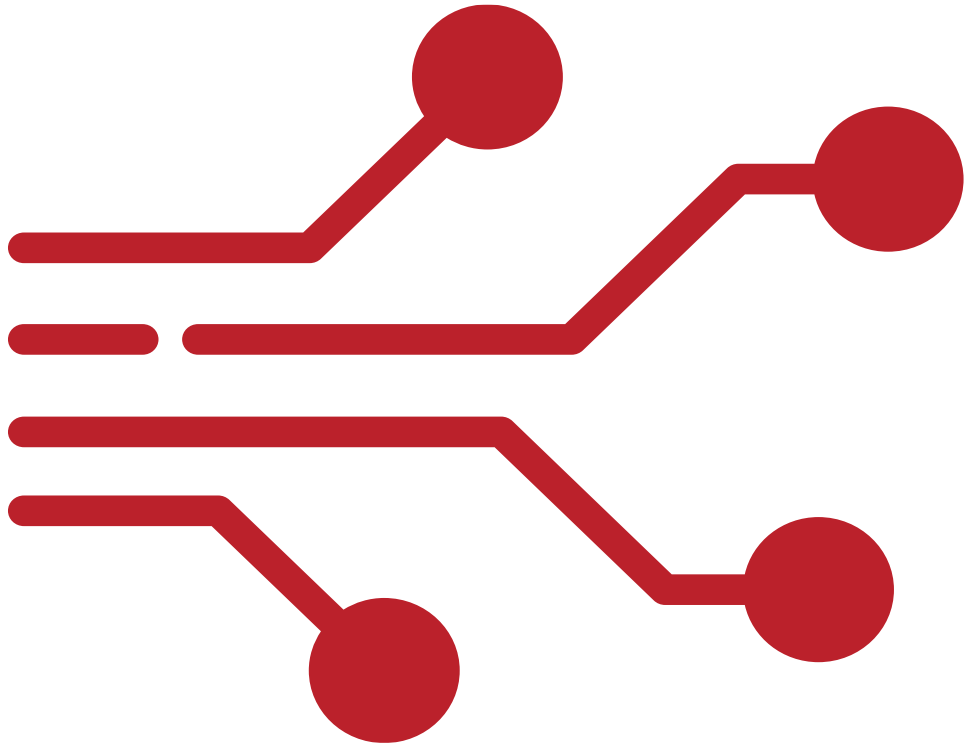
```
int F(int n)
{
    // 1) Warunek bazowy – musi się kiedyś spełnić:
    if (n == 0) return 1;

    // 2) Krok rekurencyjny – mniejszy problem:
    return n * F(n - 1);
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

194

Rekurencja w praktyce

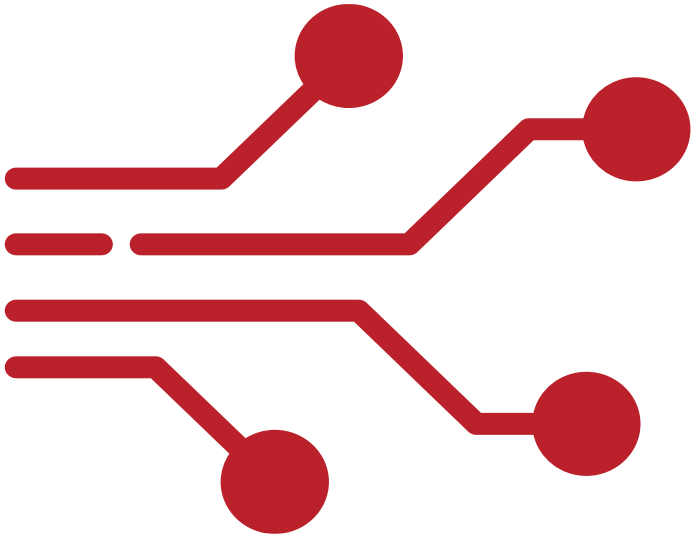


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

195

Silnia (!) — wersja rekurencyjna i iteracyjna

Definicja matematyczna:



0! = 1

n! = n * (n-1)! dla n>0

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

196

Kod rekurencyjny

```
long Faktorial(int n)
{
    if (n < 0) throw new ArgumentOutOfRangeException(nameof(n));
    if (n == 0) return 1;
    return n * Faktorial(n - 1);
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

197

Kod iteracyjny

```
long FaktorialIter(int n)
{
    if (n < 0) throw new ArgumentOutOfRangeException(nameof(n));
    long res = 1;
    for (int i = 2; i <= n; i++)
        res *= i;
    return res;
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

198

Silnia „podwójna” (!!) — implementacje

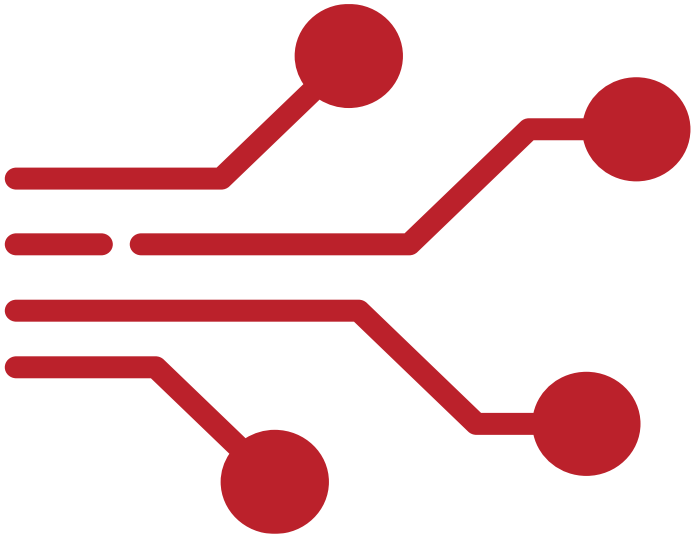
```
long PodwojnyFaktorial(int n)
{
    if (n < 0) throw new ArgumentOutOfRangeException(nameof(n));
    if (n == 0 || n == 1) return 1;
    return n * PodwojnyFaktorial(n - 2);
}
long PodwojnyFaktorialIter(int n)
{
    if (n < 0) throw new ArgumentOutOfRangeException(nameof(n));
    long res = 1;
    for (int i = n; i > 1; i -= 2) res *= i;
    return res;
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

199

Stos wywołań i bezpieczeństwo

- Każde wywołanie to nowa ramka na stosie.
- Brak domknięcia oznacza StackOverflowException.
- Wektor ataku: stack exhaustion (DoS poprzez ekstremalne zagnieżdżenie wejścia).

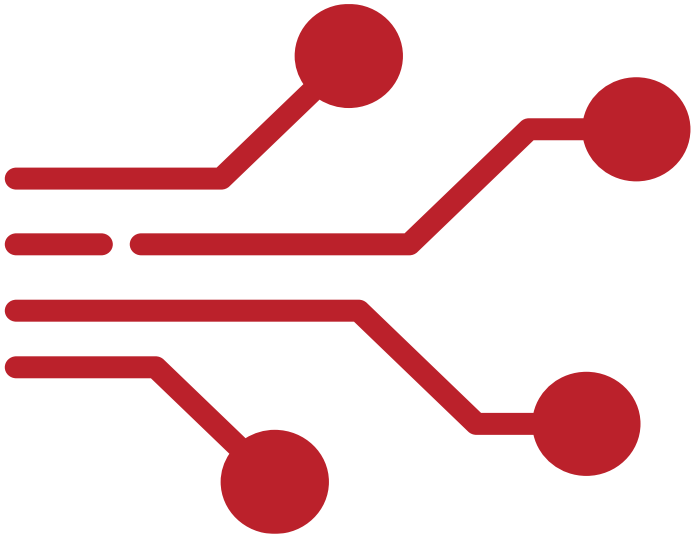


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

200

Rekurencyjny obchód katalogów



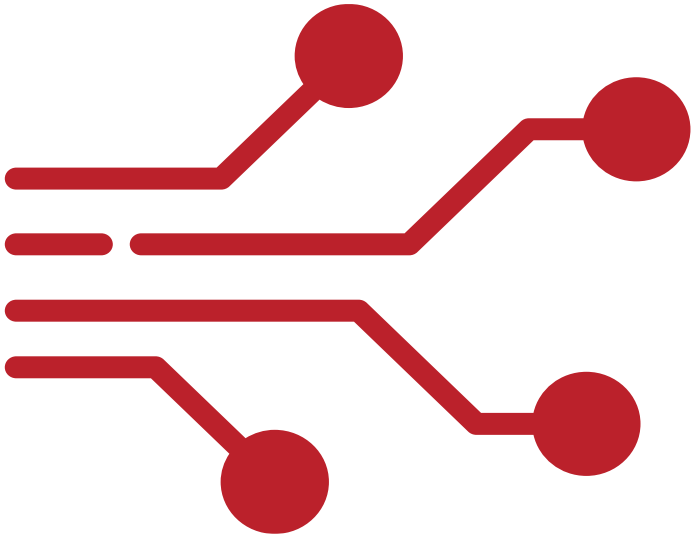
```
void ScanDir(string path)
{
    foreach (var file in Directory.GetFiles(path))
        Console.WriteLine(file);

    foreach (var dir in Directory.GetDirectories(path))
        ScanDir(dir); // rekurencja
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

201

DFS (Depth-First Search (przeszukiwanie w głąb)) w grafach (sieć, zależności)



```
void Dfs(Node v, HashSet<Node> seen)
{
    if (!seen.Add(v)) return;
       // Add zwraca false, gdy już był
    foreach (var u in v.Neighbors)
        Dfs(u, seen);
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

202

Memoizacja — przykład. Odległość Levenshteina (edytorska)

Formalnie jest to metryka w przestrzeni ciągów znaków, zdefiniowana następująco:

- działaniem prostym na napisie nazwiemy:
  - wstawienie nowego znaku do napisu
  - usunięcie znaku z napisu
  - zamianę znaku w napisie na inny znak

odległością pomiędzy dwoma napisami jest najmniejsza liczba działań prostych, przeprowadzających jeden napis na drugi.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Kod

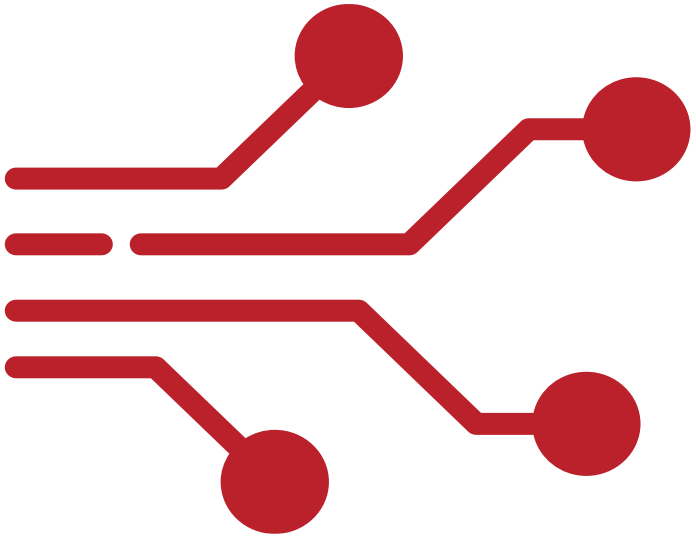
```
int Levenshtein(string a, string b)
{
    var memo = new Dictionary<(int i, int j), int>();
    return D(a.Length, b.Length);
    int D(int i, int j)
    {
        if (memo.TryGetValue((i, j), out var val)) return val;
        int ans;
        if (i == 0) ans = j;
        else if (j == 0) ans = i;
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

204

Kod

```
            else
            {
                int cost = (a[i - 1] == b[j - 1]) ? 0 : 1;
                ans = Math.Min(
                        Math.Min(D(i - 1, j) + 1, D(i, j - 1) + 1),
                        D(i - 1, j - 1) + cost
                    );
            }
            memo[(i, j)] = ans;
            return ans;
        }
    }
```



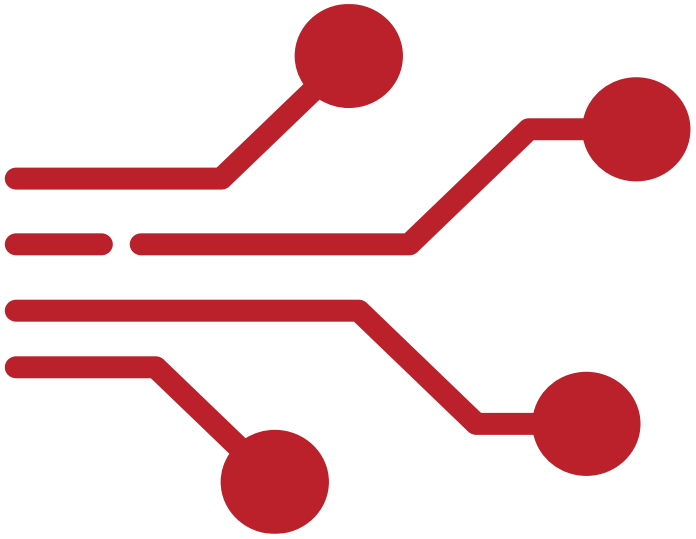
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

205

Memoizacja

- To zapamiętywanie wyników wywołań funkcji dla danych argumentów.
- Przy kolejnych wywołaniach funkcja nie liczy od nowa, tylko sprawdza w pamięci podręcznej (cache).
- Stosowana, gdy występują nakładające się podproblemy.



Schemat:

1. Wołanie funkcji z parametrami.
2. Sprawdzenie: „Czy już to liczyliśmy?”

   Jeżeli tak, to zwócenie wyniku z cache’a.

   Jeżeli nie, to obliczamy i zapisujemy do cache’a.
3. Następnym razem oszczędzamy czas.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

206

Rekurencja ogonowa — świadomość ograniczeń w C#

```
long FaktorialTail(int n, long acc = 1)
{
    if (n == 0) return acc;
    return FaktorialTail(n - 1, acc * n);
      // ostatnia instrukcja - rekurencja ogonowa
}
```

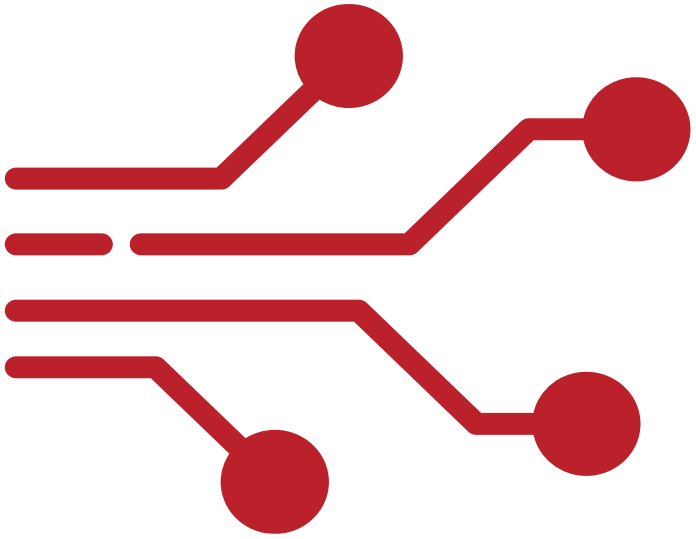
Rekurencja ogonowa to taki przypadek rekurencji, w którym ostatnią operacją w funkcji jest wywołanie tej samej funkcji (lub – szerzej – innej funkcji w łańcuchu rekurencji pośredniej). „Ostatnią” znaczy: po powrocie z tego wywołania funkcja już nic nie robi (nie ma mnożeń, dodawań, łączenia list, logowania, itp.).

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

207

Rekurencja vs iteracja — decyzja inżynierska

- Rekurencja: kod krótki, naturalny przy drzewach/grafach; ryzyko stosu.
- Iteracja: kontrola zasobów, stabilność, czasem dłuższy kod.
- Zasada: czy problem jest naturalnie rekurencyjny? Jeśli tak — OK, ale z limitami.

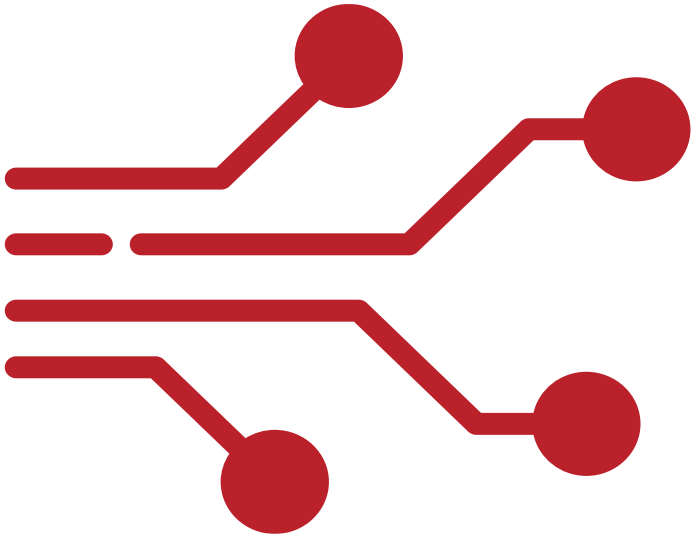


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

208

Rekurencja – podsumowanie



- Rekurencja to elegancja i bliskość definicji.
- Memoizacja to ulepszenie dla nakładających się podproblemów (np. odległość Levenshteina).
- Silnia (!) i (!!) — typowe zastosowania + ciąg Fibonacciego – do przećwiczenia.
- W C# należy świadomie zarządzać głębokością wywołań.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

209

Dziękuję za uwagę

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Wykład nr 6
Temat: Podstawowe struktury danych. Tablice. Listy

Paweł Karczmarek
Katedra Inteligencji Obliczeniowej Politechniki Lubelskiej

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Plan prezentacji

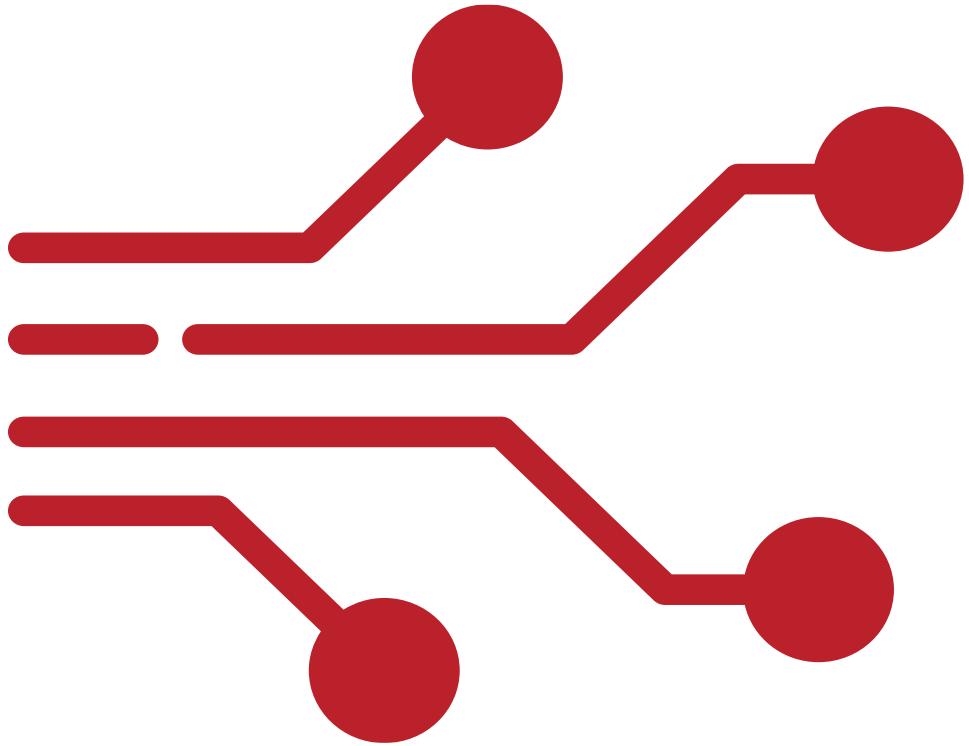
- Wprowadzenie i motywacja
- Tablice
- Listy
- Słowniki i iteratory
- Operacje CRUD i podsumowanie

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

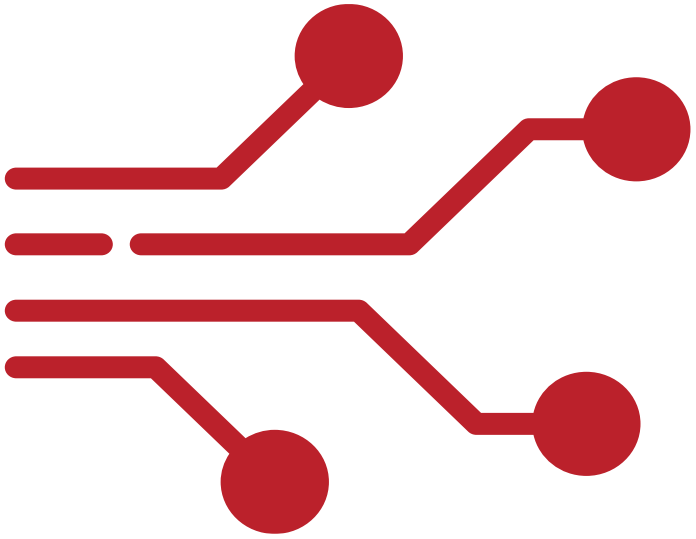
212

Wprowadzenie i motywacja



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

213

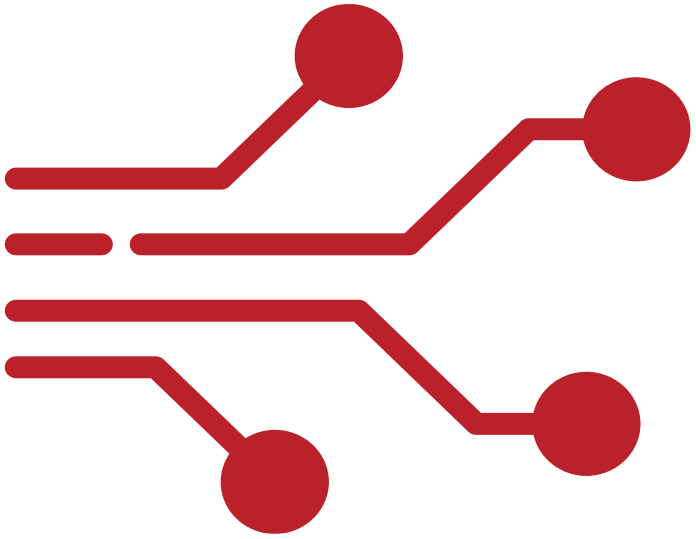


Co to jest struktura danych?

- Definicja: sposób organizacji i przechowywania danych w pamięci komputera
- Cel: szybki i efektywny dostęp, modyfikacja i zarządzanie danymi
- Przykłady: tablica, lista, stos, kolejka, słownik, graf, drzewo
- Wpływ na wydajność programów
- Zastosowania w cyberbezpieczeństwie: logi, pakiety, dane sesyjne

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Statyczne a dynamiczne struktury danych



- Tablica – rozmiar ustalany w momencie tworzenia, niezmienny
- Lista – rozmiar dynamiczny, automatyczne rozszerzanie
- Tablica: szybki dostęp do elementu przez indeks
- Lista: łatwe dodawanie i usuwanie elementów, brak konieczności znajomości rozmiaru z góry
- Wybór zależy od problemu: pamięć vs. elastyczność

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

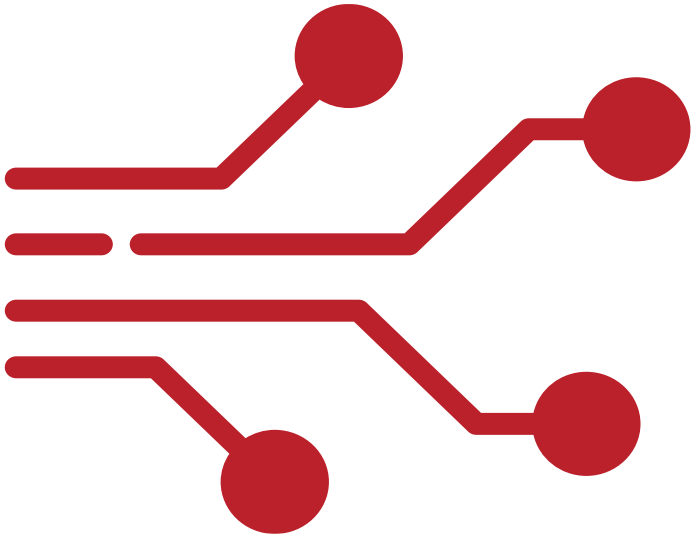
Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Tablice

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

216

Deklaracja i inicjalizacja tablic



Deklaracja referencji do tablicy

```
int[] t;
```

Deklaracja referencji wraz z utworzeniem obiektu tablicy (rezerwowana jest pamięć na stercie)

```
int[] t = new int[3]; //Domyślnie ma zera
int[] t = new int[3] { 1, 2, 4 }; //Inicj.
int[] ti = { 1, 2, 4 }; //Wariant
//Uwaga na indeksowanie elementów od 0
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

217

Tablice dwuwymiarowe

Przykład

```
int[,] t2 = new int[2, 3] { { 0, 1, 2 }, { 3, 4, 5 } };
Console.WriteLine("Liczba elementów: " + t2.Length);
Console.WriteLine("Liczba wymiarów: " + t2.Rank);
Console.WriteLine("Wielkość: " + t2.GetLength(0) + " x " +
t2.GetLength(1));
//Inicjalizacja:
int[,] ti2 = new int[2, 3];
for (int i = 0; i < 2; i++)
for (int j = 0; j < 3; j++)
ti2[i, j] = 3 * i + j;
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

218

Length i foreach

```
int[] ti = { 1, 2, 4 };
        for (int i = 0; i < ti.Length; i++)
                Console.Write("" + ti[i] + ";");

foreach (int element in ti)
        Console.Write("" + element + ";");

foreach (var i in ti2)
        Console.Write("" + i + ";");
```

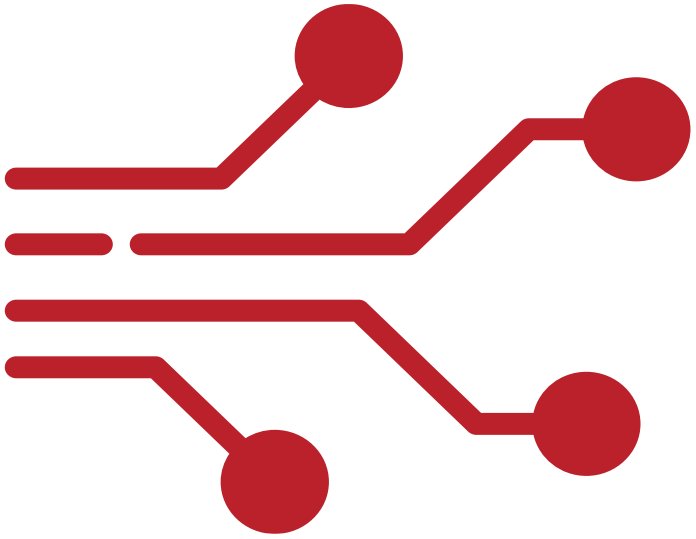

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

219

Przypisanie

```
t[2] = 9;
```

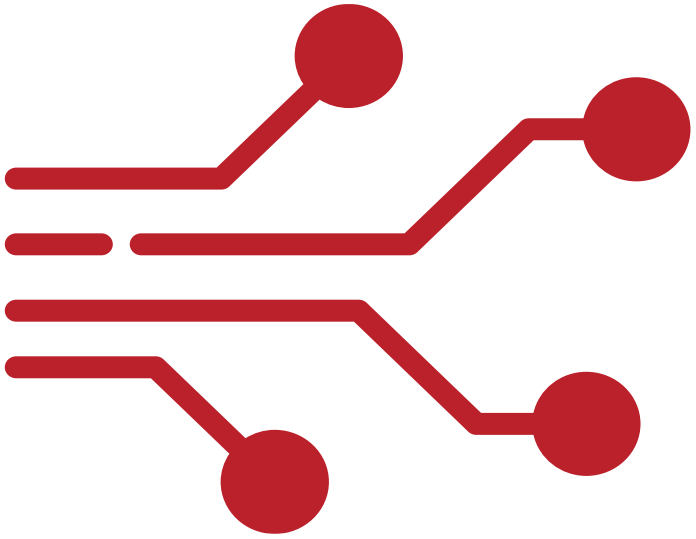


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

220

Wybór elementów

```
int[] ti = { 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 };
int[] _ti = ti[3..6]; // indeksy 3, 4, 5!
```



`ti[..3]` — elementy od początku tablicy do trzeciego (indeksy 0, 1 i 2),

`ti[3..]` — elementy od czwartego (o indeksie 3) do końca tablicy (indeksy 3, 4, 5, 6 itd.),

`ti[^1]` — ostatni element w tablicy,

`ti[^6..^3]` — od szóstego elementu od końca do trzeciego elementu od końca,

`ti[3..^3]` — od czwartego elementu do czwartego od końca.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

221

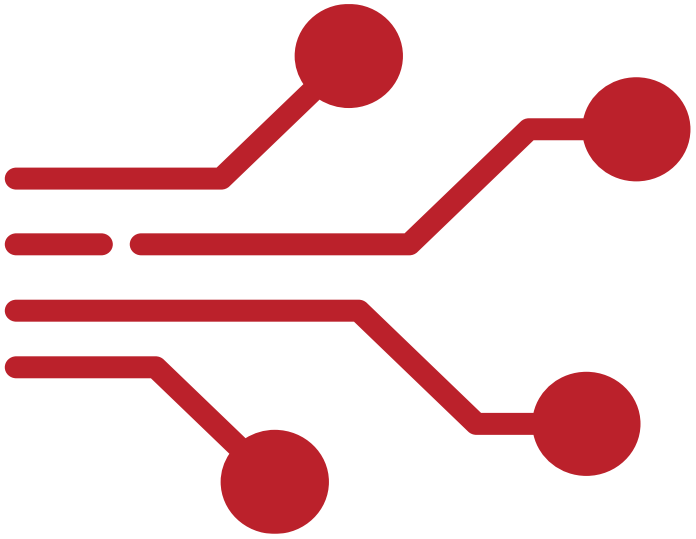
Tablice prostokątne a poszarpane

- Tablica wielowymiarowa (prostokątna):

```
int[,] macierz = new int[3,4];
```

- Tablica poszarpana (jagged array):

```
int[][] jagged = new int[3][];
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

222

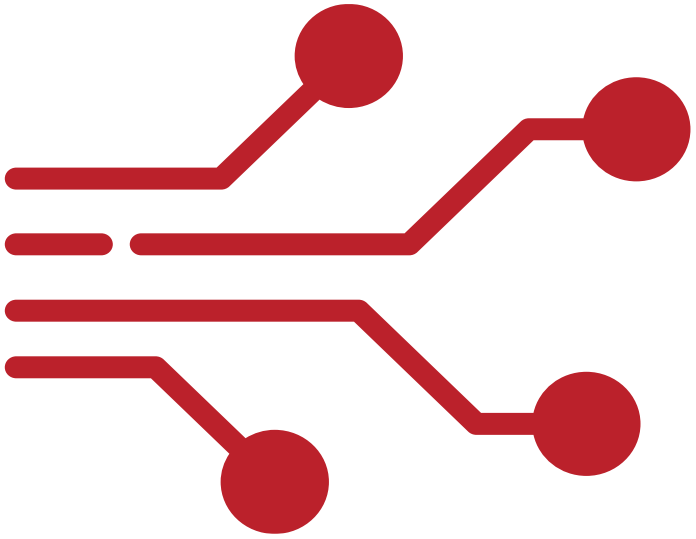
Tablice jako argumenty metod

```
static int sumuj(int[] ti)
{
    int suma = 0;
    foreach (int element in ti)
    suma += element;
    return suma;
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

223

Tablice jako argumenty metod



```
static int[] klonuj(int[] tablica)
{
        int[] kopia = new int[tablica.Length]; //tworzenie
                //nowej tablicy
        for (int i = 0; i < tablica.Length; ++i)
                kopia[i] = tablica[i]; //kopiowanie elementów
                        //z oryginalnej do nowej tablicy
        return kopia;
}
```

Uwaga: można też użyć metody Clone()

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

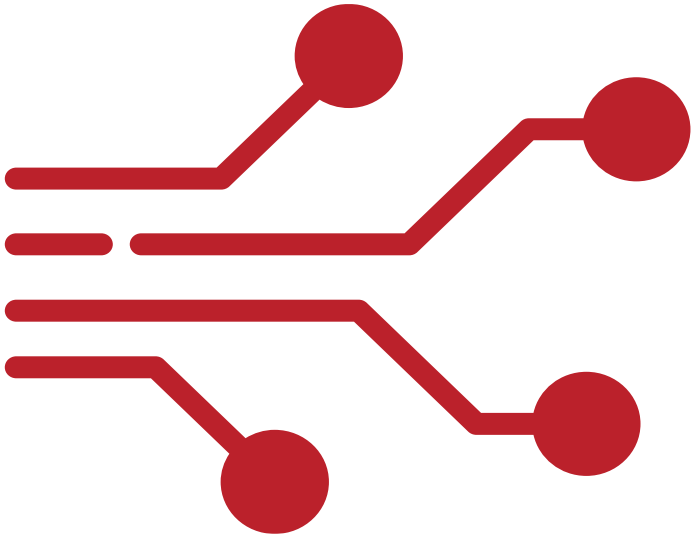
224

Sortowanie

```
int[] losy = new int[30];
Random r = new Random();
for (int indeks = 0; indeks < losy.Length; indeks++)
      losy[indeks] = r.Next(100);
string s = "Przed sortowaniem: ";
foreach (int los in losy) s += los.ToString() + " ";
Console.WriteLine(s);
Array.Sort(losy);
s = "Po sortowaniu: ";
foreach (int los in losy) s += los.ToString() + " ";
      Console.WriteLine(s);
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

225



Przeszukiwanie

- Przeszukiwanie liniowe (linear search) – sprawdzanie po kolei, O(n)
- Przeszukiwanie binarne (binary search) – tylko dla tablic posortowanych, O(log n)

Wbudowane metody:

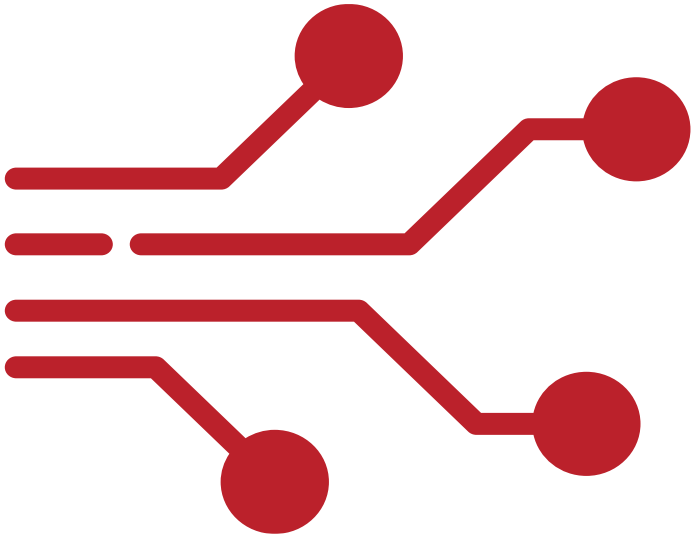
- Array.IndexOf(array, element)
- Array.LastIndexOf(...)
- Array.BinarySearch(array, element)

Wybór algorytmu to kompromis między prostotą a wydajnością

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

226



Dodawanie i usuwanie elementów

- Tablica ma stały rozmiar – nie można „dodać” ani „usunąć” elementu bezpośrednio

Typowe podejście:

- Tworzymy nową tablicę o większym rozmiarze
- Kopiujemy stare elementy
- Dodajemy nowy element na końcu

Usuwanie to kopiowanie do mniejszej tablicy

Alternatywa: używać List<T> zamiast tablic, jeśli modyfikacje są częste

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00
Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Sprawdzanie istnienia elementu w tablicy

- Array.Exists(array, predicate) – czy jest element spełniający warunek
- Array.IndexOf(array, element) – zwraca indeks lub -1
- Array.Find(array, predicate) – zwraca pierwszy element spełniający warunek
- Array.TrueForAll(array, predicate) – czy wszystkie elementy spełniają warunek
- Alternatywa: LINQ: array.Any(x => x == szukany)

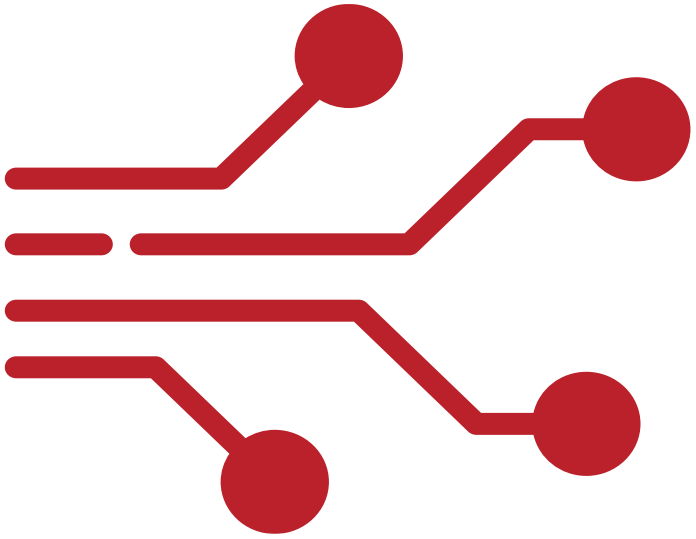
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

228

Przykład: histogram i rzut kostkami – do wykonania

- Ćwiczenie: symulacja rzutów dwiema kostkami
- Tablica liczników: int[] wyniki = new int[13];
- Pętla losująca rzuty powoduje zwiększanie licznika dla sumy oczek
- Wynik: histogram rozkładu sum od 2 do 12
- Analogia: analiza częstości zdarzeń w logach

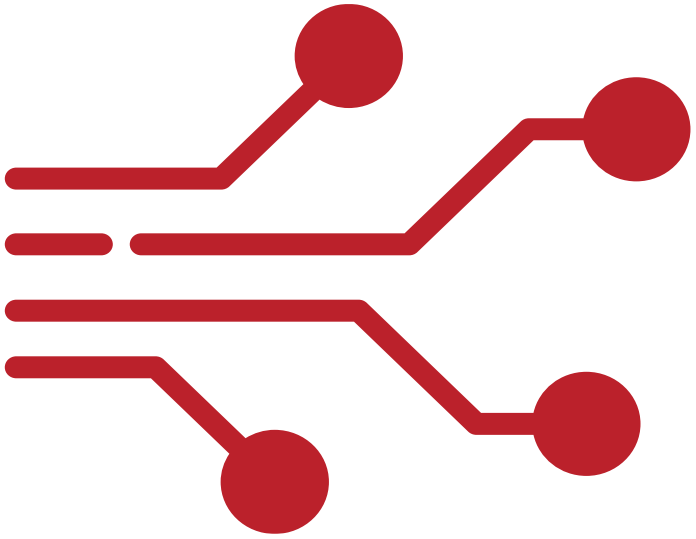


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Listy

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

230



Lista

Zgodnie z dokumentacją „reprezentuje silnie typizowaną listę obiektów, do których można uzyskać dostęp za pomocą indeksu. Udostępnia metody do wyszukiwania, sortowania i manipulowania listami.”

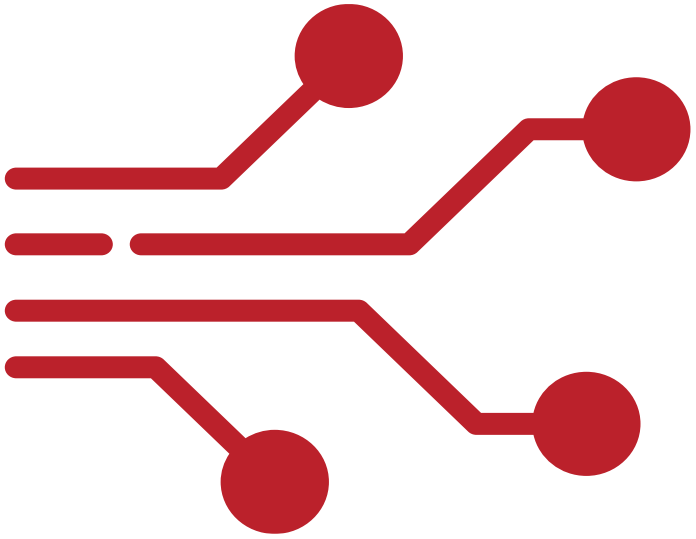
- Automatyczne zarządzanie rozmiarem
- Operacje: Add, Insert, Remove, RemoveAt
- Indeksowanie tak samo jak w tablicach

```
var lista = new List<int>();
lista.Add(10);
lista.Add(20);
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

231



Tworzenie i podstawowe metody

- Tworzenie:

```
var lista = new List<string>();
```

- Dodawanie: Add, AddRange
- Wstawianie w dowolne miejsce: Insert
- Usuwanie: Remove, RemoveAt

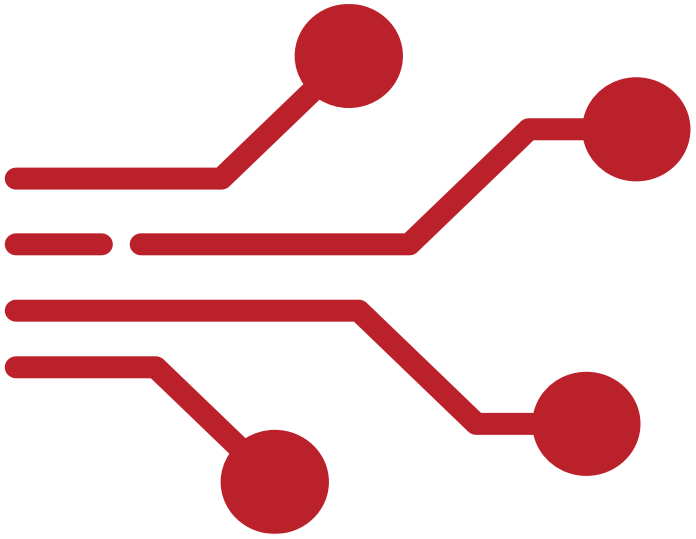
Przykład:

```
lista.Add("Ala");
lista.Insert(0, "Ola");
lista.Remove("Ala");
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

232



Indeksowanie i Count

- Dostęp jak w tablicy: lista[2]
- Rozmiar: lista.Count
- Uwaga: rozmiar zmienia się dynamicznie przy dodawaniu i usuwaniu

Przykład:

```
var l = new List<int> {1,2,3};
Console.WriteLine(l[1]); // 2
Console.WriteLine(l.Count); // 3
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Iteracja i modyfikacja listy – pułapki

Iteracja: foreach i for

- Usuwanie podczas iteracji: ryzykowne

Bezpieczne podejście:

- Iteracja od końca, np. `for (int i = l.Count-1; i >= 0; i--)`
- Tworzenie nowej listy z filtrem

Przykład:

```
foreach (var x in l)
        if (x < 0) l.Remove(x); // błąd!
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

234

Iteracja i modyfikacja listy z LINQ

LINQ pozwala filtrować listy bez bezpośredniej modyfikacji

- Bezpieczna alternatywa dla usuwania w pętli
- Tworzenie nowej listy zamiast zmiany istniejącej

Przykład:

```
List<int> liczby = new List<int>{ -3, -1, 0, 2, 5 };
var dodatnie = liczby.Where(x => x > 0).ToList();
```

- Where to wybór elementów spełniających warunek
- ToList() to materializacja wyników

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

235

LINQ

LINQ to skrót od Language Integrated Query – język zapytań wbudowany w C#

- Pozwala pisać zapytania do danych jak w SQL, ale w kodzie C#
- Działa na: tablicach, listach, kolekcjach (i nie tylko)

Podstawowe operacje:

- Where – filtruj elementy
- Select – wybierz i przekształć elementy
- OrderBy – posortuj
- ToList / ToArray – zamień na listę/tablicę

Przykład:

```
int[] liczby = {1,2,3,4,5,6};
var parzyste = liczby
               .Where(x => x % 2 == 0)
               .ToList();
```

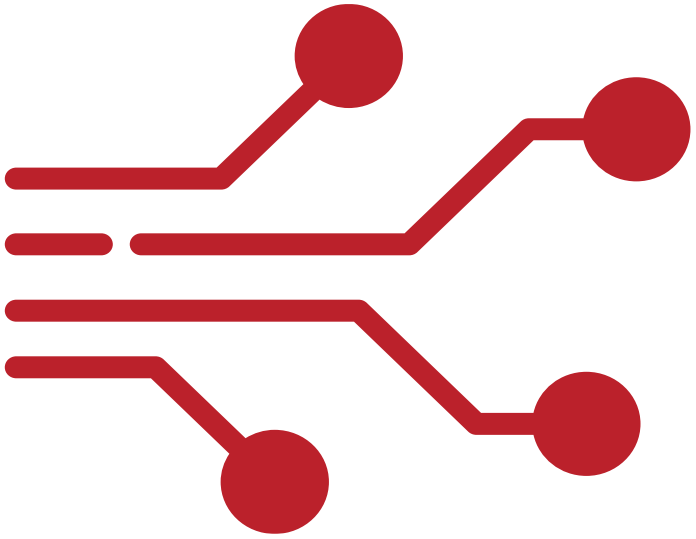

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

236

Sorotwanie listy

lista.Sort() – sortowanie in-place



Własny porównywacz:

lista.Sort((a,b)=>a.CompareTo(b));

LINQ:

lista.OrderBy(x => x) daje nową sekwencję

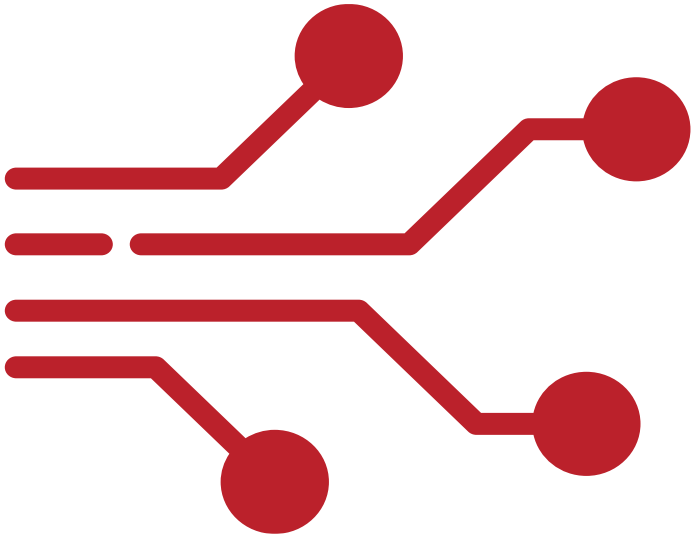
Różnica: Sort modyfikuje, OrderBy tworzy nową kolekcję IEnumerable<T> (może po niej iterować pętla foreach).

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

237

Przeszukiwanie listy



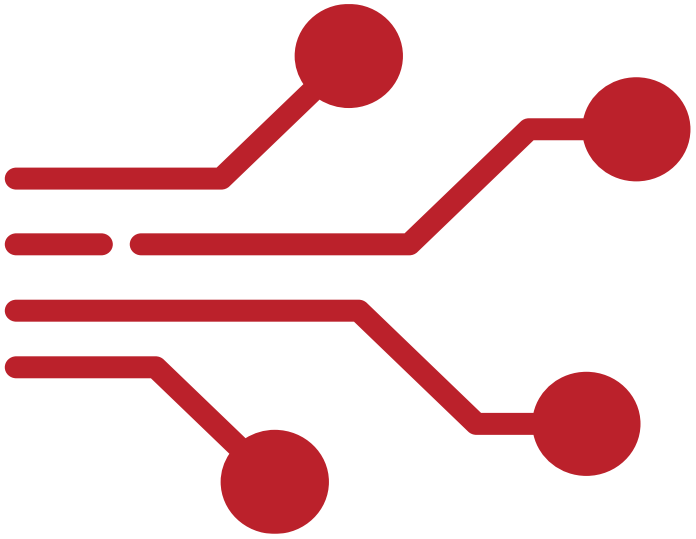
- Contains(x) – czy element istnieje
- IndexOf(x) – zwraca indeks lub -1
- Find(predicate) – pierwszy element spełniający warunek
- Exists(predicate) – czy istnieje element spełniający warunek

Złożoność: O(n) (przeszukiwanie liniowe)

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

238



Uwaga o złożoności obliczeniowej algorytmu

Złożoność oznacza jak „szybko rośnie czas wykonania” (lub użycie pamięci) w zależności od ilości danych n

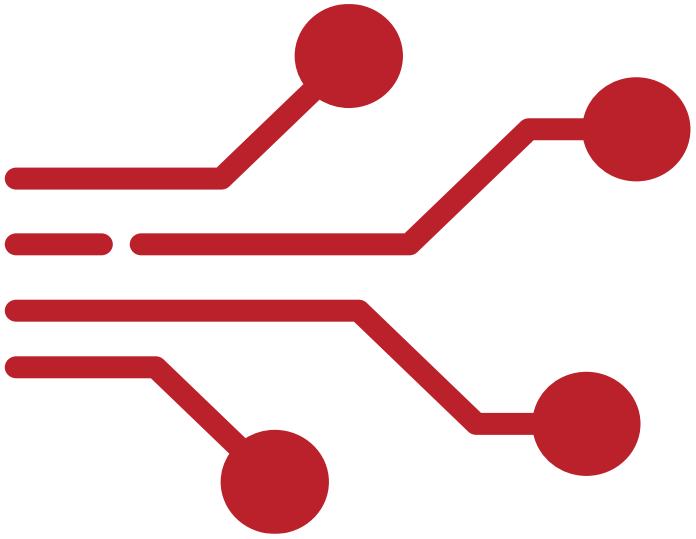
Najczęstsze rodzaje:

- $O(n)$ – liniowa: czas rośnie proporcjonalnie do liczby danych
- $O(n^2)$ – kwadratowa: czas rośnie jak n razy n (dużo wolniej przy większych danych)
- $O(\log n)$ – logarytmiczna: czas rośnie bardzo wolno przy wzroście danych

Przykłady

- $O(n)$: przeszukanie tablicy
- $O(n^2)$: porównanie każdego elementu z każdym
- $O(\log n)$: wyszukiwanie binarne w posortowanej tablicy

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00
Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską



Złożoność podstawowych operacji

- Add – amortyzowane $O(1)$
- Insert – $O(n)$ (przesunięcia elementów)
- Remove – $O(n)$ (znajdowanie elementu + przesunięcie)
- RemoveAt – $O(n)$ (przesunięcie elementów)

W praktyce: dodajemy na końcu, unikamy częstych operacji Insert/Remove w środku listy.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

240

Kopiowanie i konwersje

- lista.ToArray() – konwersja do tablicy
- Konstruktor kopiujący: new List<T>(collection)
- lista2 = new List<int>(lista); – nowa lista z tymi samymi elementami
- Płytka kopia – elementy referencyjne nadal wskazują na te same obiekty

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

241

Przykład

- Ćwiczenie: z tablicy wybierz tylko liczby parzyste i zapisz w liście
- Kod:

```
int[] liczby = {1,2,3,4,5,6};
List<int> parzyste = new List<int>();
foreach (var x in liczby)
        if (x % 2 == 0) parzyste.Add(x);
```

- Alternatywa LINQ:

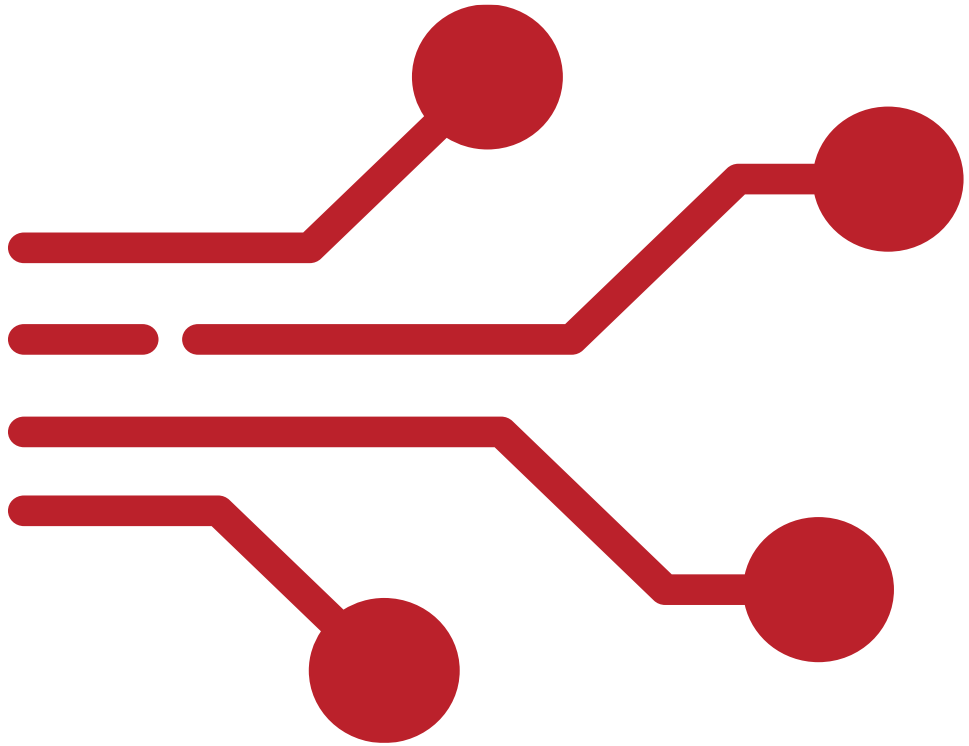
```
var parzyste = liczby.Where(x => x % 2 == 0).ToList();
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

242

Słowniki i iteratory



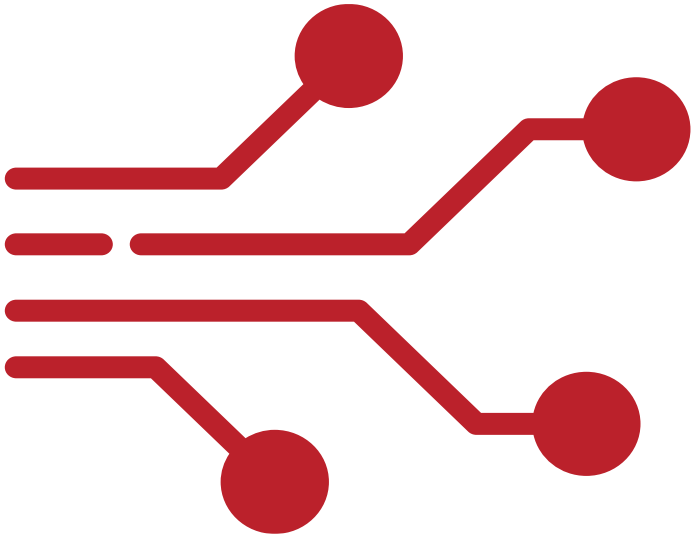
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

243

Słowniki (Dictionary<TKey,TValue>) – wprowadzenie

- Słownik to struktura klucz–wartość
- Unikalne klucze, szybki dostęp O(1)
- Tworzenie:



```
var dict = new Dictionary<string,int>();
dict["admin"] = 5;
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00
Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

244

Typowe operacje na Dictionary

- Dodawanie: dict.Add(key, value)
- Odczyt: var v = dict[key];
- Sprawdzanie: ContainsKey(key)
- Usuwanie: Remove(key)
- Bezpieczny odczyt:

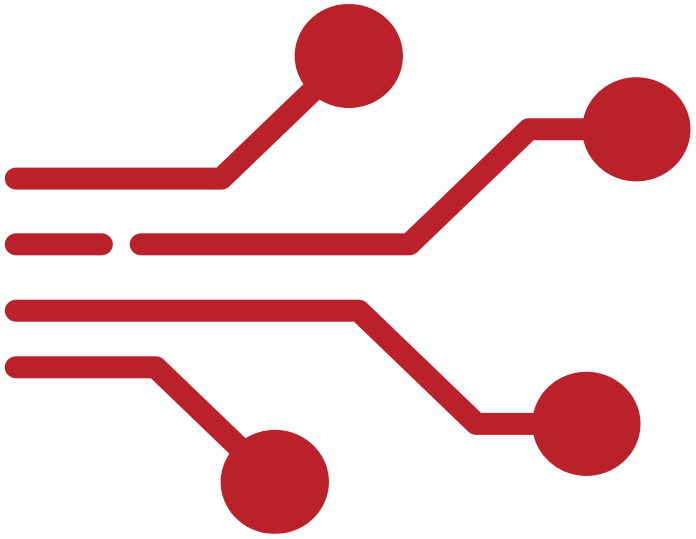
```
if (dict.TryGetValue("admin", out var val))
        ...
//ewentualnie
if (dict.ContainsKey("admin"))
    Console.WriteLine(dict["admin"]);
```

- dict["admin"] = 10; // nadpisanie istniejącej wartości

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

245



Iteratory i foreach

- Każda kolekcja implementuje IEnumerable<T>
- Możemy po niej przechodzić w pętli foreach
- Enumerator to „wskaźnik” na bieżący element

Przykład:

```
List<int> l = new List<int>{1,2,3};
foreach (var x in l) Console.WriteLine(x);
```

- foreach to bezpieczniejsza i wygodniejsza forma iteracji

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Struktury LIFO i FIFO. Stosy i kolejki – czyli jak zarządzać danymi w czasie i przestrzeni pamięci

- Stos (Stack) – Last In, First Out: ostatni dodany element zostanie pobrany jako pierwszy.
- Kolejka (Queue) – First In, First Out: pierwszy dodany element zostanie pobrany jako pierwszy.

Stos ~ talerze na stołówce (nakładamy z góry, zdejmujemy z góry).

Kolejka ~ klienci w banku (pierwszy w kolejce zostaje obsłużony pierwszy).

Zastosowanie:

- struktury algorytmiczne,
- stos wywołań funkcji,
- systemy zdarzeń i buforowania,
- analiza logów bezpieczeństwa.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

247

Stack<T>: zarządzany stos bez pułapek pamięci

```
Stack<int> stos = new Stack<int>();
stos.Push(10);
stos.Push(20);
stos.Push(30);
Console.WriteLine($"Peek: {stos.Peek()}");  // 30
Console.WriteLine($"Pop: {stos.Pop()}");    // 30
Console.WriteLine($"Pop: {stos.Pop()}");    // 20
Console.WriteLine($"Pozostało: {stos.Count}");
```

- Push() – dodaje element na wierzch stosu.
- Pop() – usuwa i zwraca element z wierzchu.
- Peek() – zwraca element bez usuwania.
- Count – liczba elementów.
- Przestrzeń nazw: System.Collections.Generic.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

248

Jak program pamięta, skąd wrócić — czyli stos wywołań funkcji i jego pułapki
Przypomnijmy silnię:

```
int Silnia(int n)
{
    if (n <= 1) return 1;
    return n * Silnia(n - 1);
}
Console.WriteLine(Silnia(5)); // 120
```

Każde wywołanie funkcji tworzy „ramkę stosową”. Ramka zawiera:

- adres powrotu,
- wartości parametrów,
- zmienne lokalne.

Po zakończeniu funkcji ramka jest usuwana. Zbyt głęboka rekurencja spowoduje StackOverflowException.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

249

Przykład. Cofanie operacji

```
Stack<string> historia = new Stack<string>();
historia.Push("Otwórz plik");
historia.Push("Edytuj zawartość");
historia.Push("Zapisz zmiany");
Console.WriteLine("Cofnięcie: " +
    historia.Pop()); // Zapisz zmiany
Console.WriteLine("Ostatnia akcja: " +
    historia.Peek()); // Edytuj zawartość
```

- Każda akcja to Push() na stos historii.
- Cofanie (Undo) to Pop().
- Podgląd bieżącego stanu to Peek().

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

250

Kolejka: przetwarzanie zdarzeń

```
Queue<string> komunikaty = new Queue<string>();
komunikaty.Enqueue("Log: Użytkownik zalogowany");
komunikaty.Enqueue("Log: Próba dostępu do pliku");
komunikaty.Enqueue("Log: Błąd autoryzacji");
Console.WriteLine(komunikaty.Dequeue()); // Log: Użytkownik zalogowany
Console.WriteLine("Następny w kolejce: " + komunikaty.Peek());
Console.WriteLine("Pozostało: " + komunikaty.Count);
```

- Enqueue() – dodaje na koniec kolejki (FIFO).
- Dequeue() – usuwa i zwraca pierwszy element.
- Peek() – podgląda pierwszy bez usuwania.
- Count – liczba elementów w kolejce.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

251

Stos vs Kolejka – porównanie i wybór właściwej struktury

Cecha: Stos / Kolejka (Queue – FIFO)

- Zasada działania: Ostatni wejdzie - pierwszy wyjdzie / Pierwszy wejdzie - pierwszy wyjdzie
- Metody kluczowe: Push(), Pop(), Peek() / Enqueue(), Dequeue(), Peek()
- Zastosowania: Cofanie operacji, rekurencja, historia zdarzeń / Kolejki zadań, logi, analiza pakietów
- Typowe ryzyka: Przepełnienie stosu - błąd StackOverflow / Zator kolejki - opóźnienia analizy
- Charakterystyka dostępu: Dostęp tylko do ostatniego elementu / Dostęp tylko do pierwszego elementu
- Przykład bezpieczeństwa: Śledzenie śladów funkcji (stack trace) / Analiza zdarzeń w chronologicznym porządku
- Złożoność operacji: *O*(1) dla Push, Pop / *O*(1) dla Enqueue/Dequeue

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

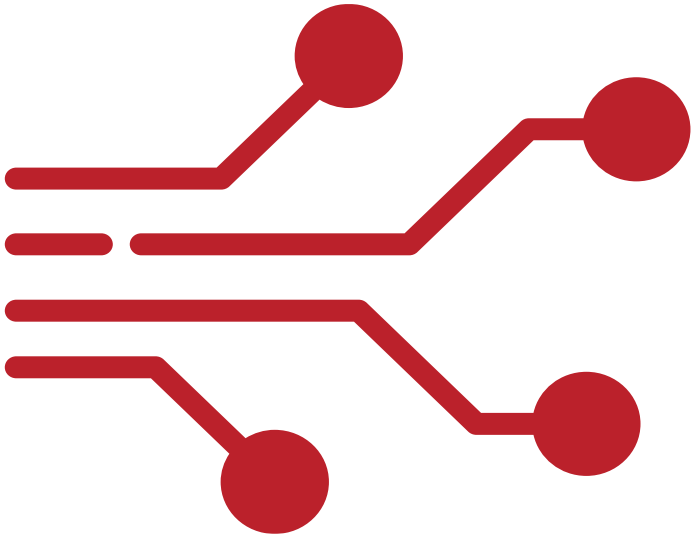
Tablice, listy i słowniki w bezpieczeństwie danych

Obszar	Struktura	Przykład zastosowania
Analiza ruchu sieciowego	Tablica	Bufor pakietów w pamięci
IDS / IPS	Lista	Kolekcja reguł detekcji lub sygnatur
Systemy uwierzytelniania	Słownik (Dictionary)	Mapowanie login - hash hasła
SIEM / logowanie	Lista	Zbiór wpisów logów przed agregacją
Kryptografia blokowe	Tablica bajtów (byte[])	Operacje XOR, szyfrowanie
Forensics	Słownik	Tłumaczenie identyfikatorów do nazw procesów
Analiza złośliwego kodu	Tablica	Obraz pamięci procesu

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

253



Stosy i kolejki w architekturze aplikacji bezpieczeństwa

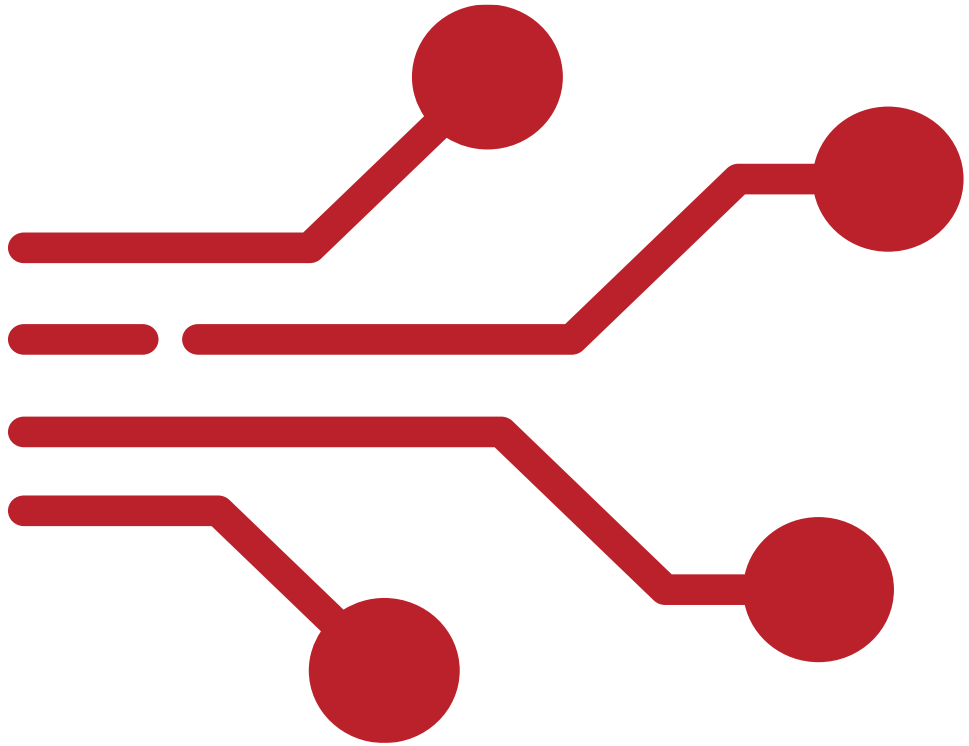
Obszar	Struktura	Przykład zastosowania
Analiza logów (SIEM)	Kolejka (FIFO)	Buforowanie zdarzeń
Analiza malware	Stos (LIFO)	Rewersja wywołań funkcji
Bazy danych	Kolejka	Rejestrowanie transakcji
Edytory kodu	Stos	Undo/Redo
Systemy kolejkowe (MQTT, Kafka)	Kolejka	Kolejkowanie pakietów
AI & ML (streamy danych)	Kolejka	Kolejka wejść dla modeli
Debugowanie	Stos	Stack trace w raportach awarii

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

254

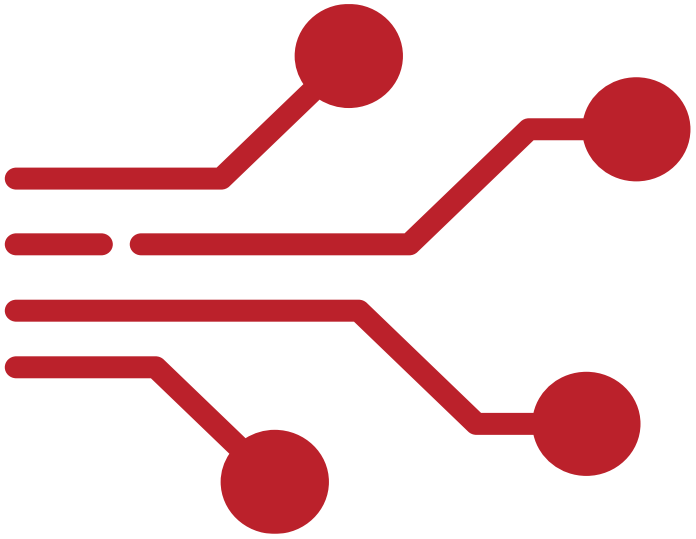
Operacje CRUD oraz podsumowanie



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

255



Operacje CRUD na wybranych kolekcjach

- CRUD = Create, Read, Update, Delete
- Create – dodanie elementu (Add)
- Read – odczyt elementu (list[i], dict[key])
- Update – zmiana elementu (list[i] = ..., dict[key] = ...)
- Delete – usuwanie (Remove)

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

256

Podsumowanie i zestawienie struktur danych

- Tablice – szybki dostęp, stały rozmiar
- Listy – dynamiczne tablice, łatwe w użyciu
- Słowniki – mapowanie klucz–wartość
- HashSet/SortedSet – unikalne elementy, szybkie sprawdzanie
- Queue/Stack – kolejki i stosy do przetwarzania danych
- SortedList/SortedDictionary – klucze posortowane

LINQ – elastyczne operacje na kolekcjach

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

257

Dziękuję za uwagę

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Wykład nr 7
Temat: Podstawy programowania obiektowego. Klasy, obiekty, właściwości, metody

Paweł Karczmarek
Katedra Inteligencji Obliczeniowej Politechniki Lubelskiej

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

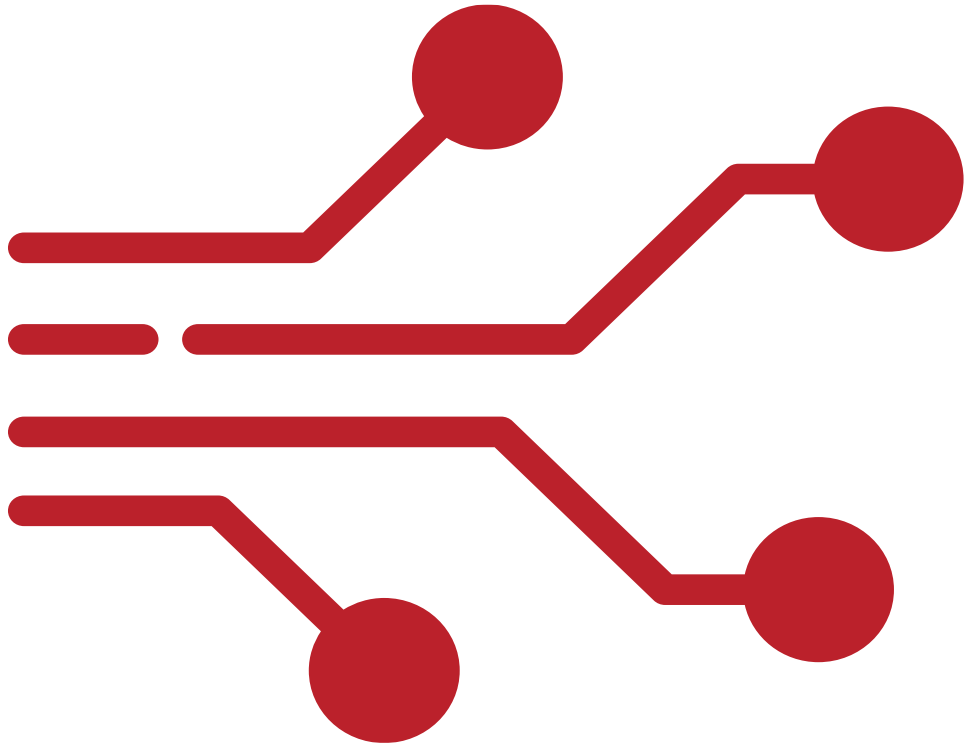
Plan prezentacji

- Wprowadzenie do obiektowości
- Klasy
- Pola i metody

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

260

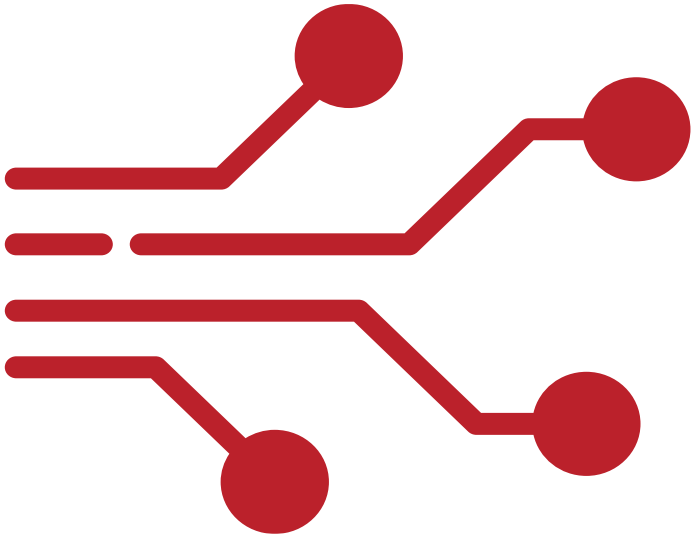
Wprowadzenie do obiektowości



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

261

Wprowadzenie do programowania obiektowego



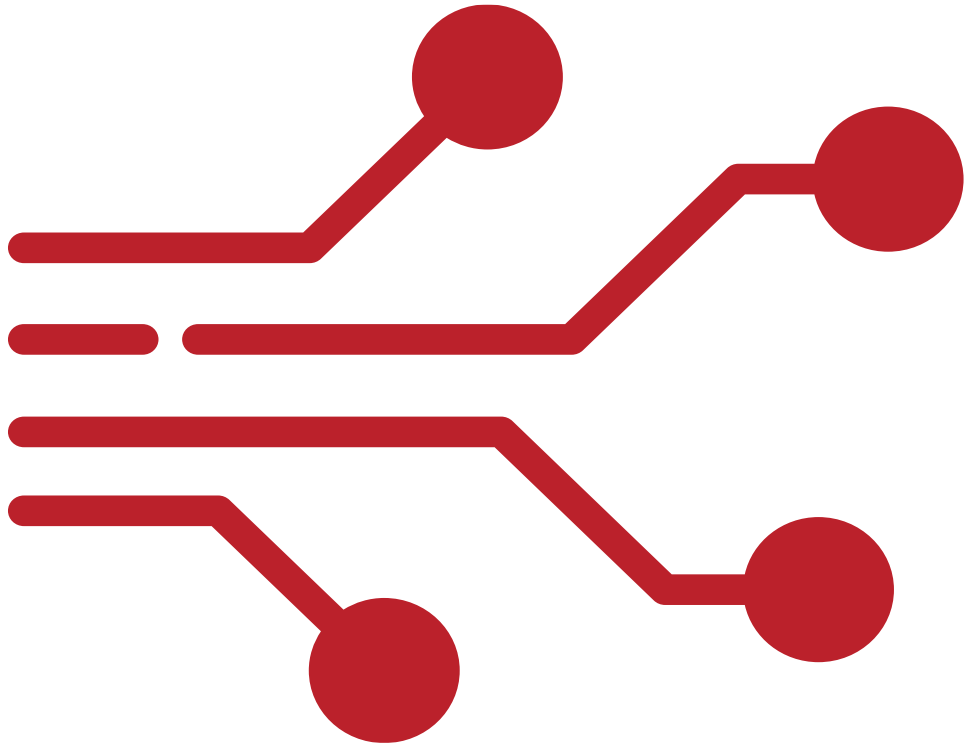
- Programowanie obiektowe (OOP – Object-Oriented Programming) to sposób tworzenia programów, w którym centrum stanowią obiekty
- Każdy obiekt łączy dane (stan) i zachowanie (metody)
- Świat programu składa się z obiektów współpracujących ze sobą
- Cel OOP: czytelność, modularność i łatwe rozszerzanie programów
- Przykład: system użytkowników, gdzie każdy obiekt Uzytkownik ma własne dane i czynności Zaloguj(), Wyloguj(), PokazProfil()

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

262

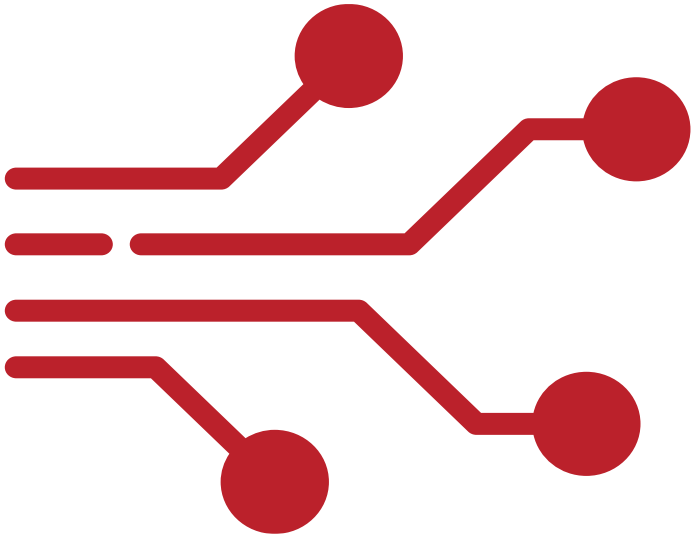
Klasy



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

263



Klasa jako szablon obiektu

- Klasa to szablon (plan, wzór) opisujący, jak mają wyglądać i zachowywać się obiekty. Określa: dane (pola i właściwości) i zachowanie (metody)
- Klasa sama nie jest obiektem – tworzy się z niej instancje
- Każdy obiekt klasy ma własne dane (np. inny login, hasło, status)
- Klasy porządkują kod i ułatwiają współpracę elementów programu

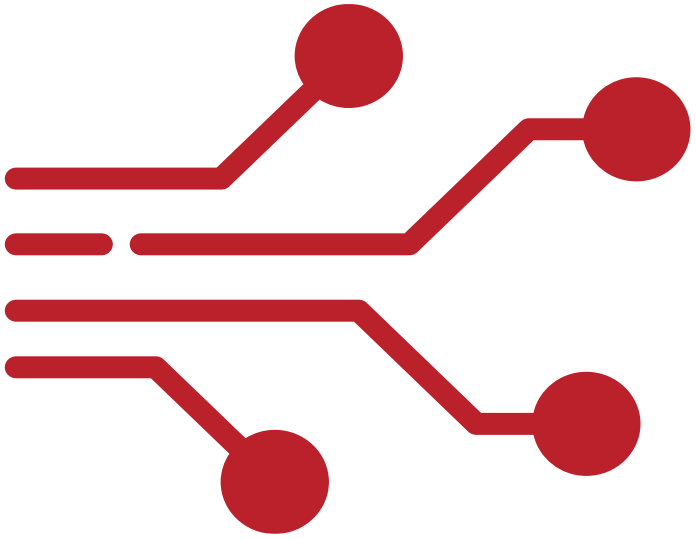
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

264

Przykład

```
class Uzytkownik
{
    public string Login; // pola (dane)
    public bool CzyZalogowany;
    public void Zaloguj() // metoda (zachowanie)
    {
        CzyZalogowany = true;
        Console.WriteLine($"{Login} został zalogowany.");
    }
    public void Wyloguj()
    {
        CzyZalogowany = false;
        Console.WriteLine($"{Login} został wylogowany.");
    }
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

265

Obiekt – konkretna instancja klasy

- Obiekt to rzeczywisty egzemplarz klasy – istnieje w pamięci programu
- Tworzy się go za pomocą słowa kluczowego new
- Każdy obiekt ma: własne dane (np. inny login), własny stan (np. zalogowany/niezalogowany)
- Obiekty tej samej klasy działają niezależnie
- Obiekt = dane + metody, które na nich działają

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

266

Przykład

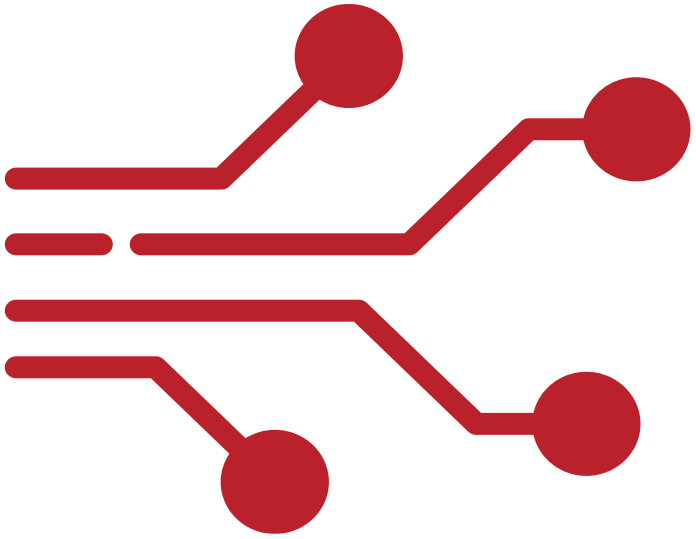
```
class Uzytkownik
{
    public string Login;
    public bool CzyZalogowany;
    public void Zaloguj()
    {
        CzyZalogowany = true;
        Console.WriteLine($"{Login} został
            zalogowany.");
    }
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

267

Przykład

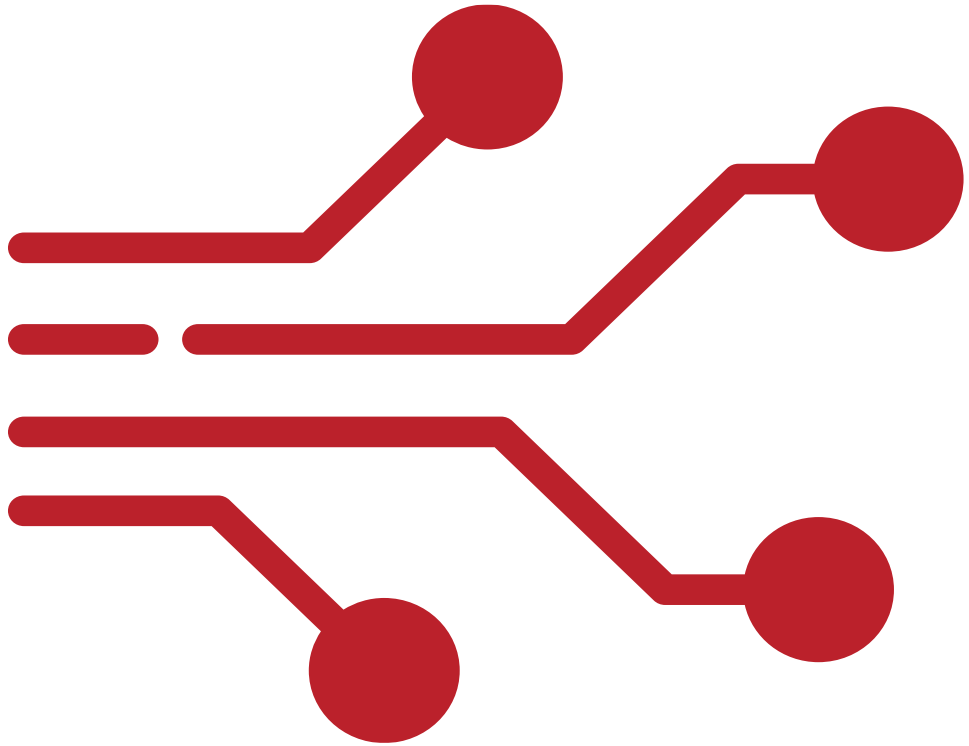


```
var u1 = new Uzytkownik();
u1.Login = "Admin";
var u2 = new Uzytkownik();
u2.Login = "Gość";
u1.Zaloguj();   // Admin został zalogowany.
u2.Zaloguj();   // Gość został zalogowany.
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

268

Pola i metody



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

269

Pola i właściwości

- Pola – przechowują dane wewnętrzne obiektu
- Właściwości (properties) – kontrolują dostęp do pól, umożliwiają odczyt (get) i zapis (set) danych w bezpieczny sposób

Dzięki właściwościom można:

- sprawdzać poprawność danych
- chronić pola przed bezpośrednią zmianą
- reagować na modyfikacje (np. logowanie zmian)

Właściwość wygląda jak pole, ale działa jak metoda

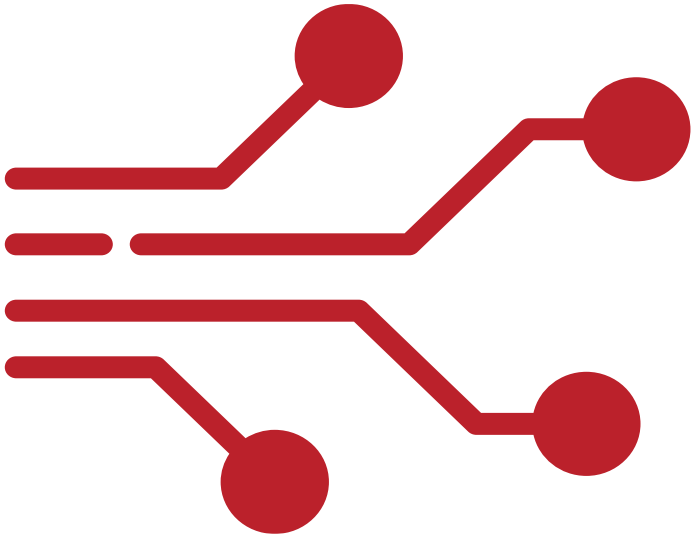
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

270

Przykład

```
class Uzytkownik
{
    // prywatne pole – dostępne tylko wewnątrz klasy
    private string _haslo;

    // publiczna właściwość – kontrolowany dostęp
    public string Login { get; set; }
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

271

Przykład

```
public string Haslo
{
    get { return "******"; } // nigdy nie zwracaj
         //prawdziwego hasła!
    set
    {
        if (value.Length >= 6)
            _haslo = value;
        else
            Console.WriteLine("Hasło musi mieć
                co najmniej 6 znaków.");
    }
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

272

Metody – zachowanie obiektu

Metoda to czynność, którą obiekt potrafi wykonać. Składa się z:

- nazwy
- listy parametrów
- typu zwracanego (lub void – gdy nic nie zwraca)

Metoda może zmieniać stan obiektu lub zwracać wynik obliczeń. Metody organizują zachowanie obiektu – co obiekt robi i jak reaguje.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

273

Przykład

```
class Uzytkownik
{
    public string Login { get; set; }
    public bool CzyZalogowany { get; private set; }

    // Metoda bez zwracania wartości
    public void Zaloguj()
    {
        CzyZalogowany = true;
        Console.WriteLine($"{Login} został zalogowany.");
    }
```

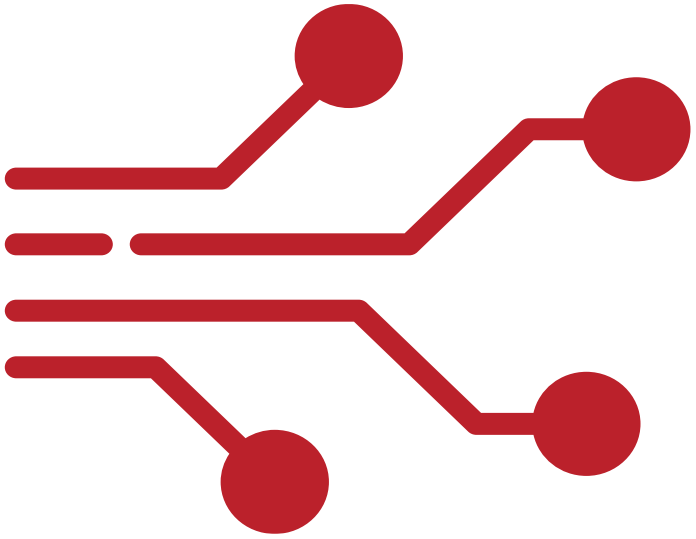

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

274

Przykład

```
// Metoda z parametrem i wartością zwracaną
    public bool SprawdzHaslo(string wpisaneHaslo,
                             string poprawneHaslo)
    {
        return wpisaneHaslo == poprawneHaslo;
    }

    public void Wyloguj()
    {
        CzyZalogowany = false;
        Console.WriteLine($"{Login} został wylogowany.");
    }
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

275

Przykład – klasa Uzytkownik

```
class Uzytkownik
{
    // Właściwości
    public string Login { get; set; }
    public bool CzyZalogowany { get; private set; }

    // Metoda logowania
    public void Zaloguj()
    {
        CzyZalogowany = true;
        Console.WriteLine($"{Login} został zalogowany.");
    }
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

276

Przykład – klasa Uzytkownik

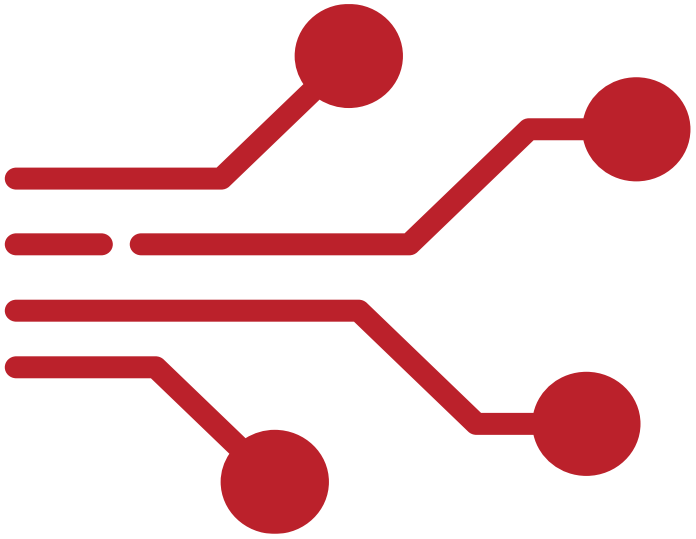
```
    // Metoda wylogowania
    public void Wyloguj()
    {
        CzyZalogowany = false;
        Console.WriteLine($"{Login} został wylogowany.");
    }

    // Metoda pomocnicza
    public void PokazStatus()
    {
        Console.WriteLine($"Użytkownik: {Login}, zalogowany: {CzyZalogowany}");
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Przykład – klasa Uzytkownik

```
        }
}
```



```
var admin = new Uzytkownik { Login = "Admin" };
var gosc = new Uzytkownik { Login = "Gość" };
admin.Zaloguj();
gosc.Zaloguj();
admin.PokazStatus();
gosc.PokazStatus();
admin.Wyloguj();
admin.PokazStatus();
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

278

Konstruktor – tworzenie i inicjalizacja obiektów

- Konstruktor to specjalna metoda wywoływana automatycznie przy tworzeniu obiektu (new)
- Ma taką samą nazwę jak klasa i nie zwraca żadnej wartości
- Służy do ustawiania początkowych wartości pól, przygotowania obiektu do działania
- Klasa może mieć kilka konstruktorów (tzw. przeciążanie konstruktorów)
- Jeśli nie zdefiniujemy konstruktora – C# doda pusty konstruktor domyślny

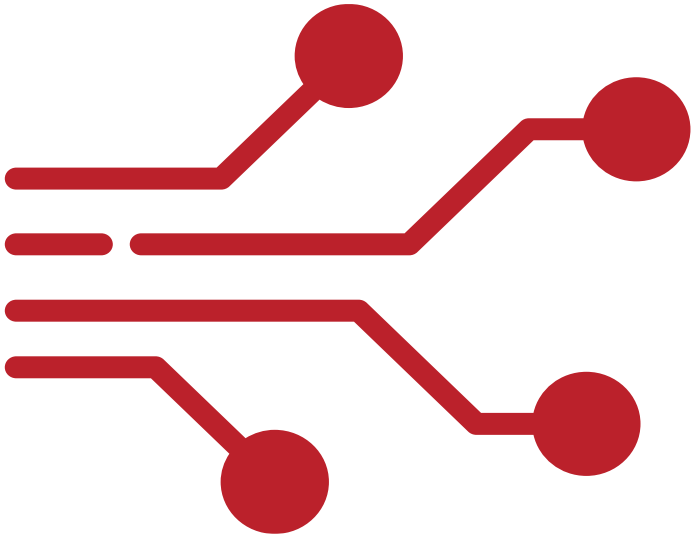
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

279

Przykład

```
class Uzytkownik
{
    public string Login { get; set; }
    public bool CzyZalogowany { get; private set; }
    // Konstruktor z parametrem
    public Uzytkownik(string login)
    {
        Login = login;
        CzyZalogowany = false;
        Console.WriteLine($"Utworzono nowego użytkownika:
            {Login}");
    }
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Przykład

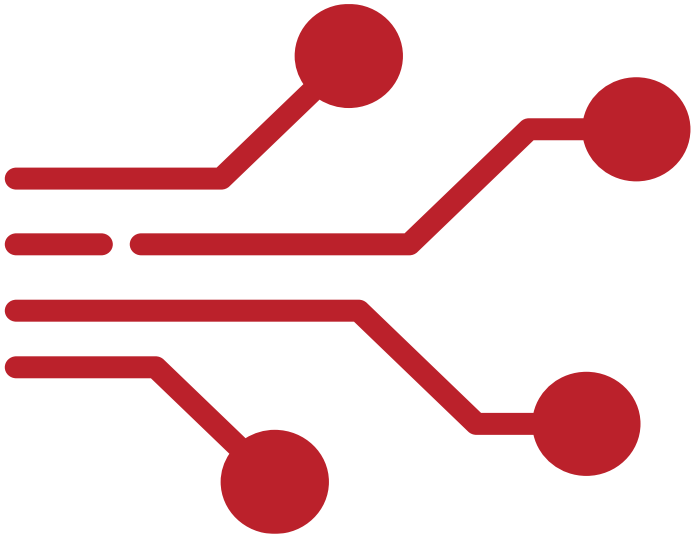
```
    public void Zaloguj()
    {
        CzyZalogowany = true;
        Console.WriteLine($"{Login} został zalogowany.");
    }
}
```

```
var admin = new Uzytkownik("Admin");
var gosc = new Uzytkownik("Gość");
admin.Zaloguj();
gosc.Zaloguj();
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

281



Słowo kluczowe this

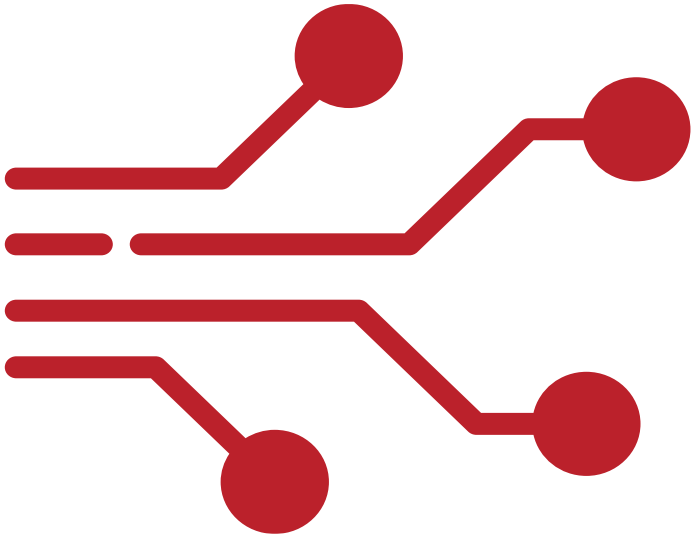
- this oznacza bieżący obiekt, czyli ten, na którym wywoływana jest metoda
- Umożliwia:
  - rozróżnienie między polem a parametrem o tej samej nazwie
  - przekazanie obiektu do innych metod lub konstruktorów
  - pisanie bardziej czytelnego kodu
- Występuje tylko w metodach niestatycznych (bo odnosi się do konkretnego obiektu)

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00
Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

282

Przykład

```
class Uzytkownik
{
    public string Login { get; set; }
    private string _haslo;
    // Konstruktor z parametrem o tej samej nazwie co pole
    public Uzytkownik(string login, string haslo)
    {
        this.Login = login;   // "this" odnosi się do pola klasy
        this._haslo = haslo;  // rozróżnia parametr od pola
    }
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

283

Przykład

```
public void PokazDane()
{
    Console.WriteLine($"Login: {this.Login}");
}


public void Porownaj(Uzytkownik inny)
{
    if (this.Login == inny.Login)
        Console.WriteLine("Obaj użytkownicy mają ten sam
           login.");
    else
        Console.WriteLine("Loginy są różne.");
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Przeciążanie metod – Overloading

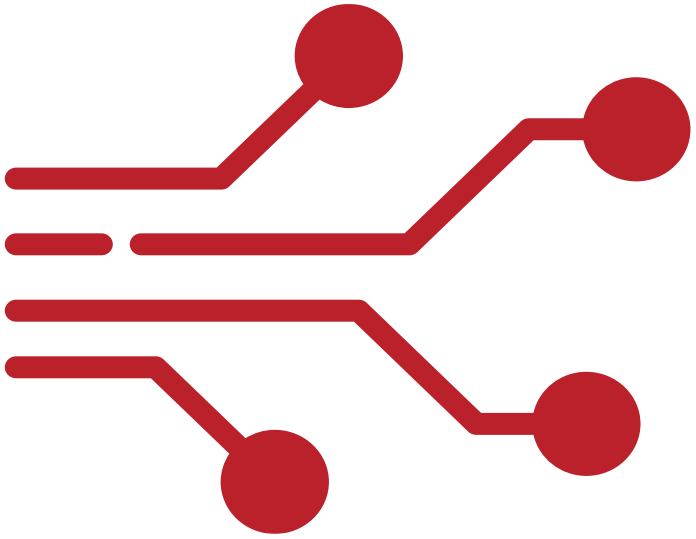
- Przeciążanie metod – możliwość zdefiniowania wielu metod o tej samej nazwie
- Różnią się liczbą, typem lub kolejnością parametrów
- Pozwala wykonywać tę samą czynność na różne sposoby
- Ułatwia tworzenie elastycznych i czytelnych interfejsów
- Nie zależy od typu zwracanego (tylko od parametrów!)

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

285

Przykład

```
class Uzytkownik
{
    public string Login { get; set; }
    public bool CzyZalogowany { get; private set; }

    // 1. wersja bez parametrów
    public void Zaloguj()
    {
        CzyZalogowany = true;
        Console.WriteLine($"{Login} zalogowany (bez podania hasła).");
    }
```



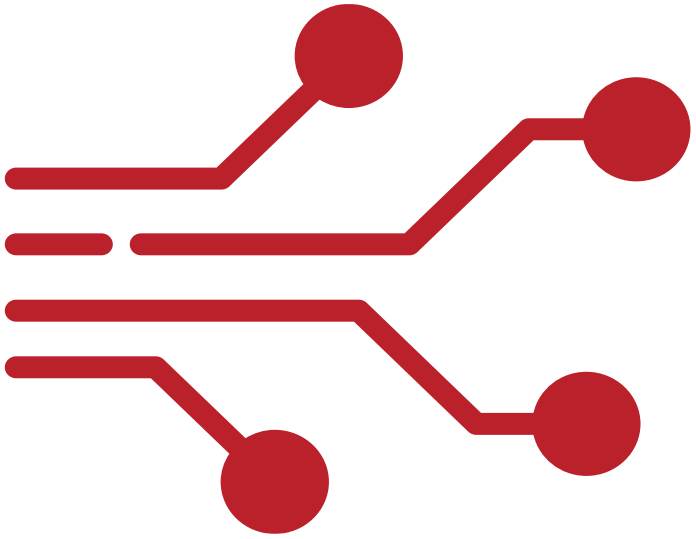
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

286

Przykład

```
// 2. wersja z parametrami
public void Zaloguj(string haslo)
{
    if (haslo == "1234") // przykładowe hasło
    {
        CzyZalogowany = true;
        Console.WriteLine($"{Login} zalogowany poprawnie.");
    }
    else
    {
        Console.WriteLine("Błędne hasło!");
    }
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

287

Przykład

```
    // 3. wersja z tokenem (np. logowanie przez aplikację)
    public void Zaloguj(string token, bool przezToken)
    {
        if (przezToken)
            Console.WriteLine($"{Login} zalogowany za pomocą tokenu:
{token}");
    }
}
var admin = new Uzytkownik { Login = "Admin" };

admin.Zaloguj();
admin.Zaloguj("1234");
admin.Zaloguj("ABC-TOKEN-123", true);
```

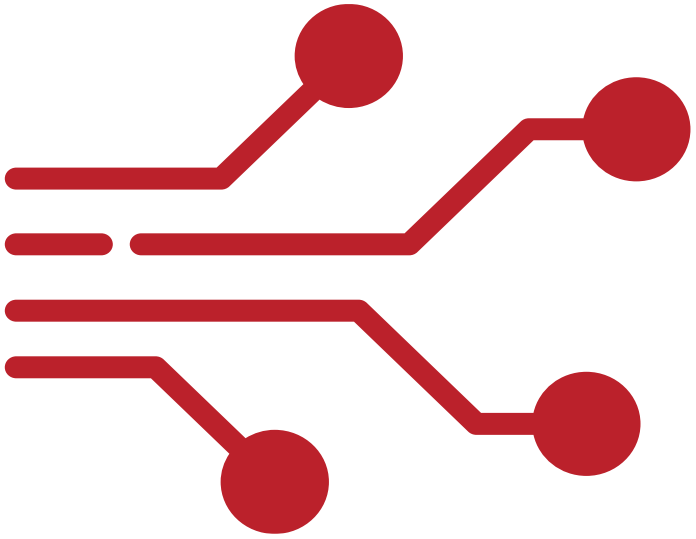

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

288

Modyfikatory dostępu

- Modyfikatory dostępu określają, kto ma prawo korzystać z elementów klasy
- Służą do ukrywania szczegółów i zabezpieczania danych



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

289

Modyfikatory dostępu

```
class Uzytkownik
{
    // dostęp tylko w tej klasie
    private string _haslo;

    // dostęp publiczny – widoczny wszędzie
    public string Login { get; set; }

    // tylko klasa i jej potomne mają dostęp
    protected bool CzyAktywny = true;
```

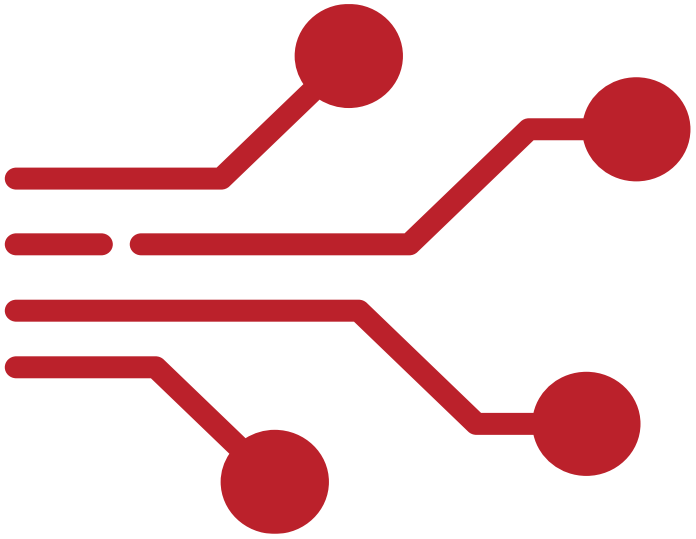

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

290

Modyfikatory dostępu

```
    // metoda widoczna publicznie
    public void PokazLogin()
    {
        Console.WriteLine($"Login: {Login}");
    }

    // metoda pomocnicza – ukryta przed światem
    private void SzyfrujHaslo()
    {
        Console.WriteLine("(Hasło zostało zaszyfrowane)");
    }
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

291

Metody statyczne i niestatyczne

- Metoda niestatyczna działa na konkretnym obiekcie (instancji klasy)
- Metoda statyczna należy do klasy, a nie do konkretnego obiektu
- Metody statyczne wywołujemy przez nazwę klasy, nie obiektu
- Metody niestatyczne mogą korzystać z this, statyczne nie
- Statyczne są wygodne, ale trzeba używać ich ostrożnie, ponieważ nie przechowują stanu

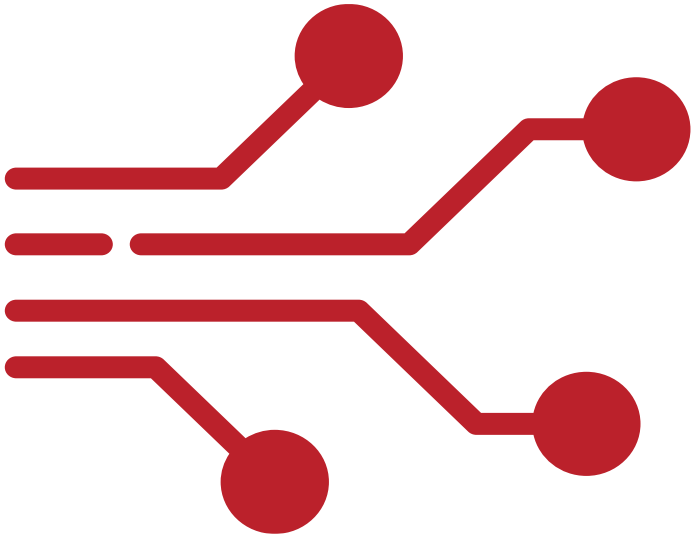
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

292

Przykład

```
class Logger
{
    // Metoda statyczna – nie wymaga tworzenia obiektu
    public static void Zapisz(string wiadomosc)
    {
        Console.WriteLine($"[LOG] {DateTime.Now}: {wiadomosc}");
    }
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

293

Przykład

```
class Uzytkownik
{
    public string Login { get; set; }

    // Metoda niestatyczna – działa na obiekcie
    public void Zaloguj()
    {
        Logger.Zapisz($"{Login} zalogował się do systemu.");
    }
}
var admin = new Uzytkownik { Login = "Admin" };
admin.Zaloguj();                    // wywołanie metody obiektowej
Logger.Zapisz("System uruchomiony."); // wywołanie metody klasowej
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

294

Kolekcje i inicjalizatory obiektów

- Inicjalizator obiektu umożliwia szybkie tworzenie i wypełnianie obiektu danymi
- Kolekcje (List, Dictionary, itp.) służą do przechowywania wielu obiektów. Dzięki nim można:
  - tworzyć listy użytkowników, logów, zdarzeń
  - przetwarzać je w pętli
- Składnia inicjalizatora:

```
var obiekt = new Klasa { Pole1 = wartość, Pole2 = wartość };
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

295

Przykład

```
class Uzytkownik
{
    public string Login { get; set; }
    public bool CzyZalogowany { get; set; }

    public void PokazStatus()
    {
        Console.WriteLine($"{Login} – zalogowany: {CzyZalogowany}");
    }
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

296

Przykład

```
class Program
{
    static void Main()
    {
        // Tworzenie listy użytkowników przy użyciu inicjalizatorów
        var uzytkownicy = new List<Uzytkownik>
        {
            new Uzytkownik { Login = "Admin", CzyZalogowany = true },
            new Uzytkownik { Login = "Gość", CzyZalogowany = false },
            new Uzytkownik { Login = "Operator", CzyZalogowany = true }
        };
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

297

Przykład

```
            // Iteracja po liście
            foreach (var u in uzytkownicy)
            {
                u.PokazStatus();
            }
        }
    }
```

Wynik:

```
Admin – zalogowany: True
Gość – zalogowany: False
Operator – zalogowany: True
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00
Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

298

LINQ – filtrowanie i przetwarzanie obiektów

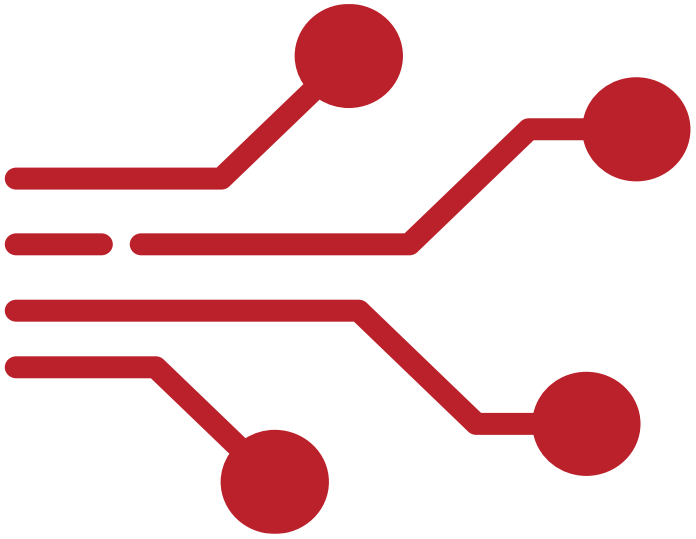
```
class Uzytkownik
{
    public string Login { get; set; }
    public bool CzyZalogowany { get; set; }
    public int LiczbaNieudanychProb { get; set; }
}

class Program
{
    static void Main()
    {
        var uzytkownicy = new List<Uzytkownik>
        {
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

299

LINQ – filtrowanie i przetwarzanie obiektów

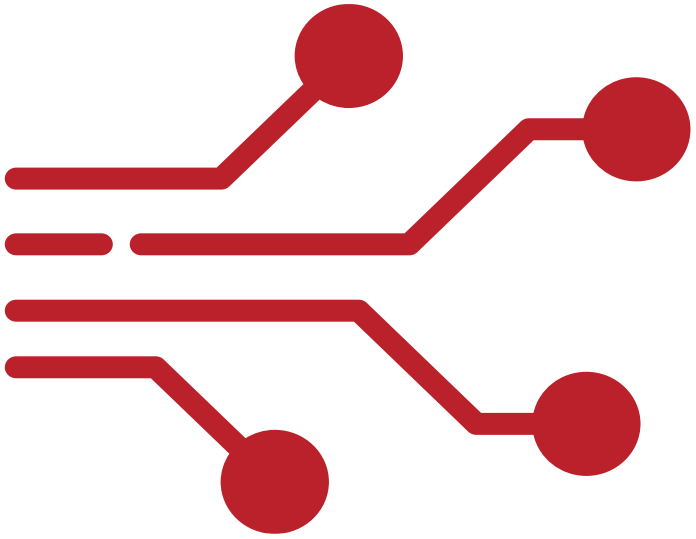


```
        new Uzytkownik { Login = "Admin", CzyZalogowany = true,
            LiczbaNieudanychProb = 1 },
        new Uzytkownik { Login = "Gość", CzyZalogowany = false,
            LiczbaNieudanychProb = 4 },
        new Uzytkownik { Login = "Operator", CzyZalogowany = true,
            LiczbaNieudanychProb = 0 },
        new Uzytkownik { Login = "Tester", CzyZalogowany = false,
            LiczbaNieudanychProb = 3 }
    };


    // Filtr: tylko niezalogowani z więcej niż 2 nieudanymi próbami
    var podejrzani = uzytkownicy
        .Where(u => !u.CzyZalogowany && u.LiczbaNieudanychProb > 2)
        .OrderBy(u => u.Login);
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

```
        Console.WriteLine("Użytkownicy z podejrzaną aktywnością:");
        foreach (var u in podejrzani)
            Console.WriteLine($"{u.Login} – {u.LiczbaNieudanychProb}
               nieudanych prób logowania");
    }
}
```



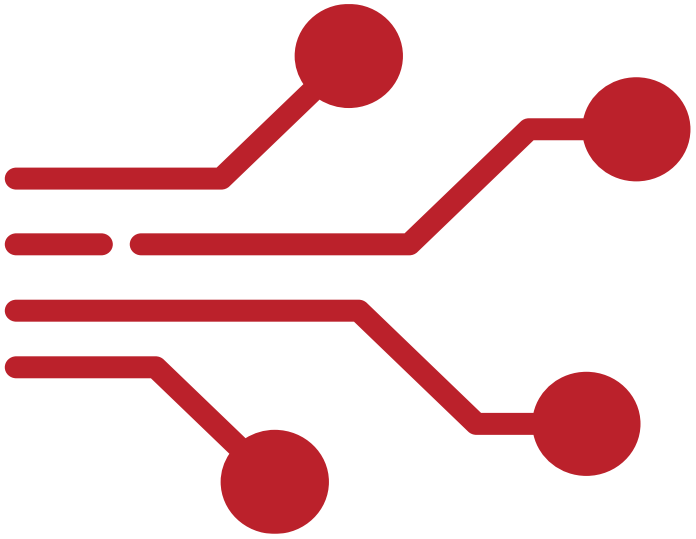
Wynik:

```
Użytkownicy z podejrzaną aktywnością:
Gość – 4 nieudanych prób logowania
Tester – 3 nieudanych prób logowania
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

301



Klasy a struktury (class a struct)

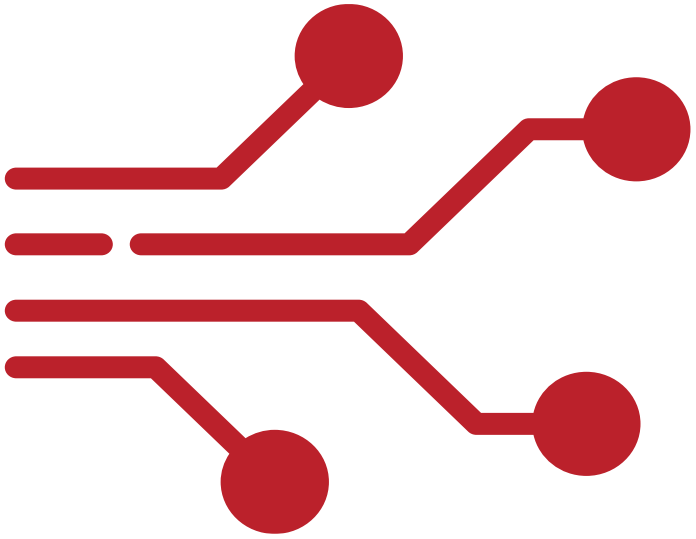
- Klasy to złożone typy obiektowe przechowywane w pamięci sterty (heap)
- Struktury to lekkie typy wartościowe przechowywane na stosie (stack)
- Oba mogą mieć pola, właściwości, metody i konstruktory
- Klasy są przekazywane przez referencję, struktury przez wartość
- Klasy mogą dziedziczyć, struktury – nie
- Struktury stosujemy do prostych danych (np. współrzędne, czas, rozmiar), można je inicjalizować bez new

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Przykład

```
class Uzytkownik
{
    public string Login { get; set; }
    public bool CzyZalogowany { get; set; }

    public void Pokaz()
    {
        Console.WriteLine($"{Login} – zalogowany: {CzyZalogowany}");
    }
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

303

Przykład

```
struct Punkt
{
    public int X;
    public int Y;
    public Punkt(int x, int y)
    {
        X = x;
        Y = y;
    }
    public void Pokaz()
    {
        Console.WriteLine($"Punkt: ({X}, {Y})");
    }
}
```

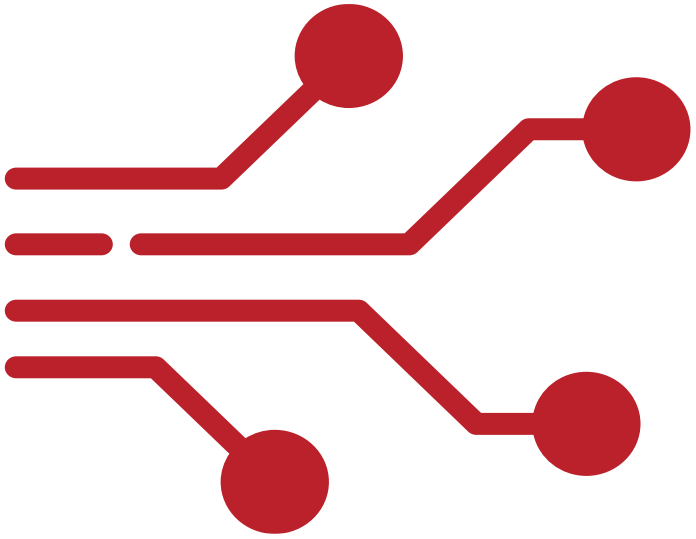

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

304

Przykład

```
var user = new Uzytkownik { Login = "Admin", CzyZalogowany = true };
user.Pokaz();
```



```
var p1 = new Punkt(10, 20);
var p2 = p1;   // kopia wartości
p2.X = 99;

p1.Pokaz(); // Punkt: (10, 20)
p2.Pokaz(); // Punkt: (99, 20)
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

305

Dziękuję za uwagę

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Wykład nr 8
Temat: Dziedziczenie. Polimorfizm

Paweł Karczmarek
Katedra Inteligencji Obliczeniowej Politechniki Lubelskiej

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Plan prezentacji

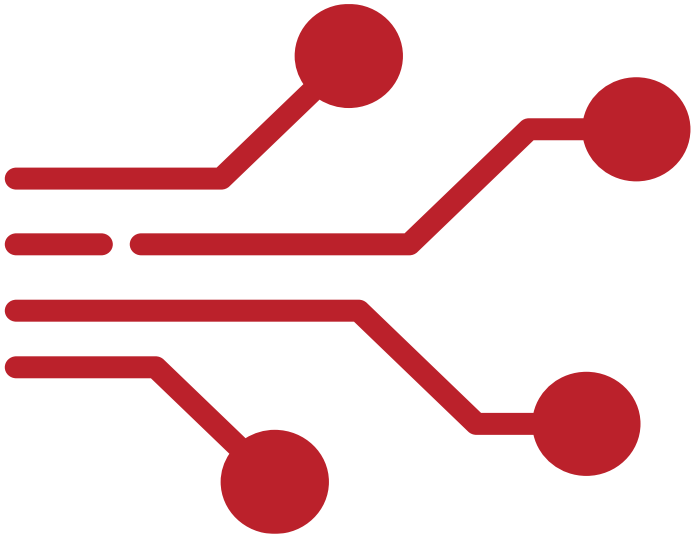
- Dziedziczenie
- Polimorfizm

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

308

Dziedziczenie

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00



Wprowadzenie – po co nam dziedziczenie?

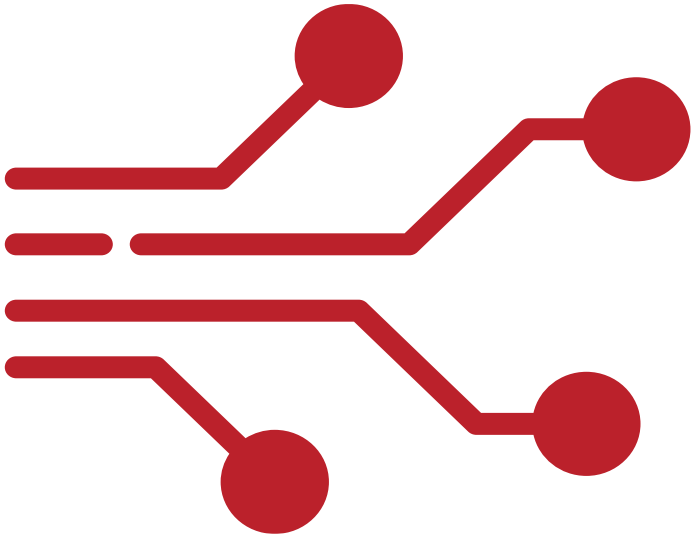
- Dziedziczenie to mechanizm przekazywania cech i zachowań z klasy bazowej do pochodnej
- Pozwala ponownie wykorzystywać kod i rozszerzać go bez duplikacji
- Klasa pochodna „dziedziczy” pola, właściwości i metody klasy bazowej
- Związek „is-a” (jest typem tego samego rodzaju)
- Przykład: Administrator jest Użytkownikiem, ale ma więcej uprawnień
- Cel: tworzenie hierarchii klas, czyli struktur odzwierciedlających świat rzeczywisty

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Przykład

```
// Klasa bazowa
class Uzytkownik
{
    public string Login { get; set; }
    public void Zaloguj()
    {
        Console.WriteLine($"{Login} zalogował się
                                do systemu.");
    }
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

311

Przykład

```
// Klasa pochodna
class Administrator : Uzytkownik
{
    public void ZmienHaslo()
    {
        Console.WriteLine($"{Login} zmienił hasło
            użytkownika.");
    }
}
var admin = new Administrator(); admin.Login = "root";
admin.Zaloguj();      // metoda odziedziczona
admin.ZmienHaslo();   // metoda własna
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

312

Podstawy składni dziedziczenia

- Dziedziczenie w C# zapisujemy przez dwukropek :
- Klasa pochodna przejmuje wszystko, co public i protected z klasy bazowej
- Składnia:

```
class KlasaPochodna : KlasaBazowa
```

- Klasa pochodna może mieć swoje własne pola, właściwości i metody
- Słowo kluczowe base pozwala odwołać się do elementów klasy bazowej
- Każda klasa (poza object) ma dokładnie jedną klasę bazową

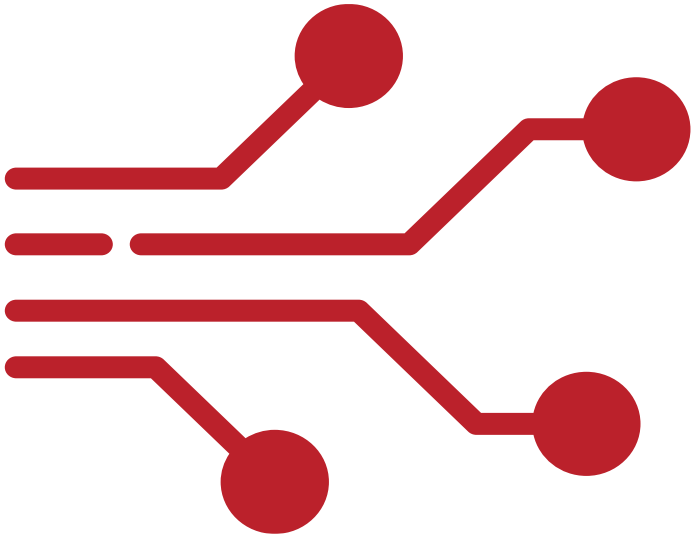
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

313

Przykład

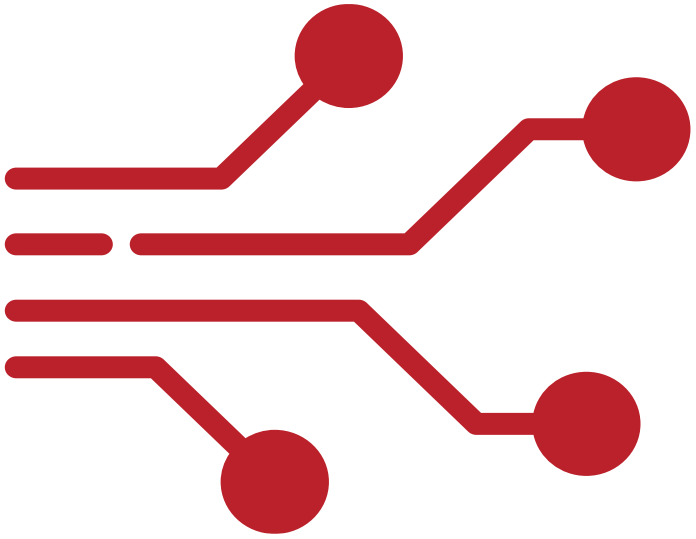
```
class Uzytkownik
{
    public string Login { get; set; }

    public void Zaloguj()
    {
        Console.WriteLine($"{Login} zalogował się do systemu.");
    }
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Przykład

```
// Klasa pochodna
class Administrator : Uzytkownik
{
    public void ZmienHaslo()
    {
        Console.WriteLine($"{Login} zmienił hasło innego
            użytkownika.");
    }
}
```

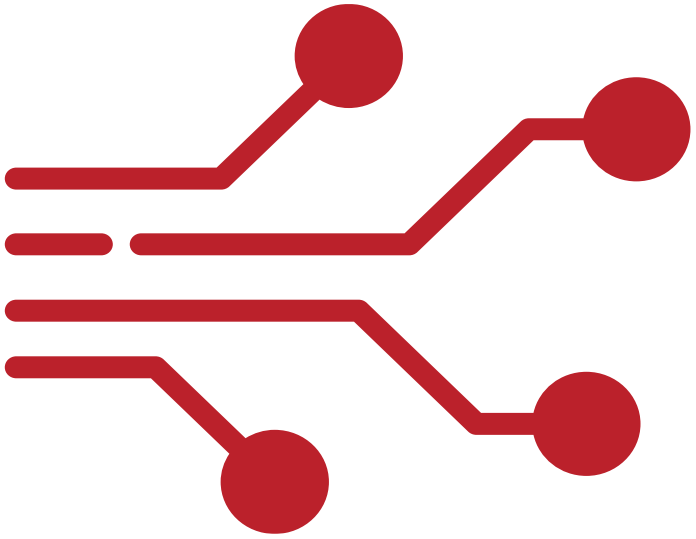


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

315

Przykład

```
// Kolejna klasa pochodna
class SuperAdministrator : Administrator
{
    public void RestartujSerwer()
    {
        Console.WriteLine($"{Login} zrestartował serwer!");
    }
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

316

Przykład

```
class Program
{
    static void Main()
    {
        var admin = new Administrator { Login = "root" };
        admin.Zaloguj();        // odziedziczona metoda
        admin.ZmienHaslo();     // własna metoda

        var super = new SuperAdministrator { Login = "sysroot" };
        super.Zaloguj();        // z klasy bazowej
        super.RestartujSerwer(); // nowa metoda
    }
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

317

Przykład

```
//Wynik działania

root zalogował się do systemu.
root zmienił hasło innego użytkownika.
sysroot zalogował się do systemu.
sysroot zrestartował serwer!
```

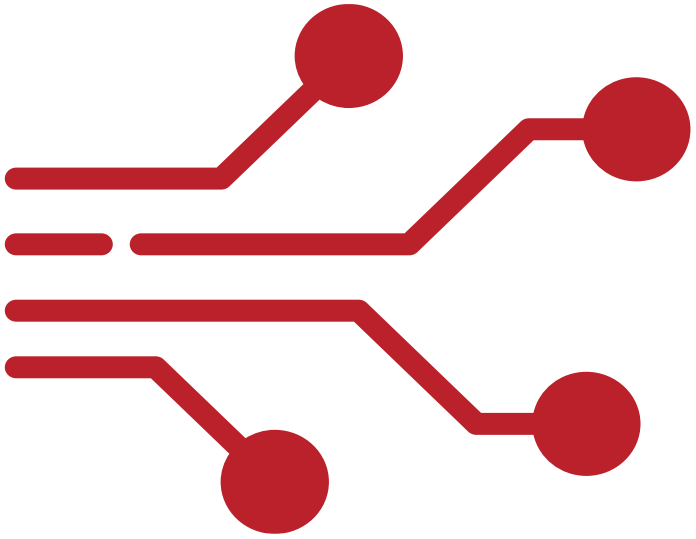

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

318

Konstruktor klasy bazowej a pochodnej

- Każda klasa może mieć swój konstruktor
- Gdy klasa dziedziczy po innej, konstruktor klasy bazowej jest wywoływany automatycznie



Kolejność:

1. Konstruktor klasy bazowej
2. Konstruktor klasy pochodnej

- Słowo kluczowe base() – jawne wywołanie konstruktora klasy bazowej
- Słowo this() – wywołanie innego konstruktora z tej samej klasy.
- Dzięki temu możemy inicjalizować część wspólną i rozszerzoną oddzielnie

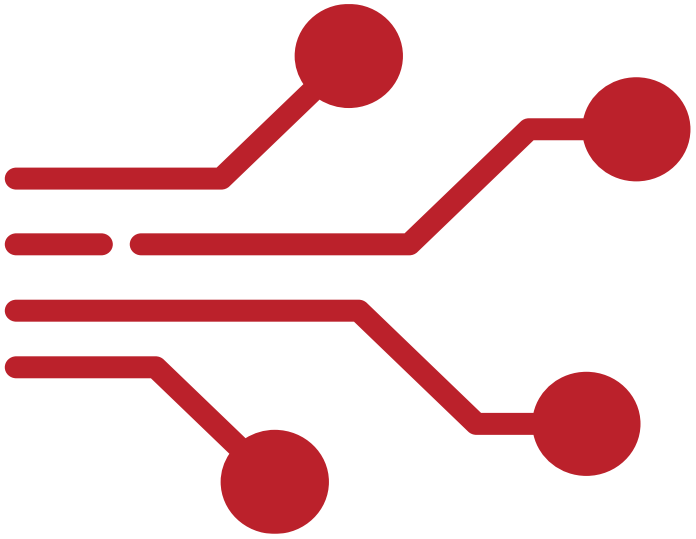
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

319

Przykład

```
class Uzytkownik
{
    public string Login { get; set; }
    // Konstruktor klasy bazowej
    public Uzytkownik(string login)
    {
        Login = login;
        Console.WriteLine($"[Uzytkownik] Utworzono konto: {Login}");
    }
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

320

Przykład

```
class Administrator : Uzytkownik
{
    public string Rola { get; set; }

    // Konstruktor pochodny wywołuje bazowy
    public Administrator(string login, string rola)
        : base(login)   // wywołanie konstruktora Uzytkownik
    {
        Rola = rola;
        Console.WriteLine($"[Administrator] Nadano rolę: {Rola}");
    }
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Przykład

```
class Program
{
    static void Main()
    {
        var admin = new Administrator("root", "SuperAdmin");
    }
}

//Wynik:
[Uzytkownik] Utworzono konto: root
[Administrator] Nadano rolę: SuperAdmin
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

322

Modyfikator protected – dane dla potomków

- protected oznacza „chroniony” – dostępny tylko dla tej klasy i jej potomnych
- Umożliwia dziedziczącym klasom korzystanie z danych bazowych
- Chroni dane przed dostępem z zewnątrz (lepsze bezpieczeństwo niż public)
- Często używany w połączeniu z private do tworzenia bezpiecznych modeli danych
- W klasie pochodnej można korzystać z pól protected tak, jakby były własne

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

323

Przykład

```
class Uzytkownik
{
    // chronione pole – nie jest publiczne, ale widoczne dla klas
       //pochodnych
    protected string Haslo;
    public string Login { get; set; }
    public void UstawHaslo(string haslo)
    { Haslo = haslo; }
    public void Zaloguj()
    { Console.WriteLine($"{Login} zalogował się do systemu.");}
}
```



```
class Administrator : Uzytkownik
```

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00
Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Przykład

```
class Administrator : Uzytkownik
{
    public void PokazHaslo()
    {
        // dostęp do chronionego pola z klasy bazowej
        Console.WriteLine($"[Admin] Widzę hasło użytkownika: {Haslo}");
    }
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

325

Przykład

```
class Program
{
    static void Main()
    {
        var admin = new Administrator { Login = "root" };
        admin.UstawHaslo("tajneprzezpoufne");
        admin.Zaloguj();
        admin.PokazHaslo();
    }
} //Wynik:
root zalogował się do systemu.
[Admin] Widzę hasło użytkownika: tajneprzezpoufne
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

326

Hierarchia klas

- Klasy można łączyć w hierarchie – od ogólnych do szczegółowych
- Klasa bazowa definiuje wspólne cechy i zachowania
- Klasy pochodne dodają własne pola i metody
- Wszystkie pochodne dziedziczą zachowania klasy bazowej
- Dzięki temu powstaje logiczna struktura podobna do systemu ról

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

327

Przykład

```
class Uzytkownik
{
    public string Login { get; set; }
    public void Zaloguj()
    {
        Console.WriteLine($"{Login} zalogował się do systemu.");
    }
    public void Wyloguj()
    {
        Console.WriteLine($"{Login} wylogował się z systemu.");
    }
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

328

Przykład

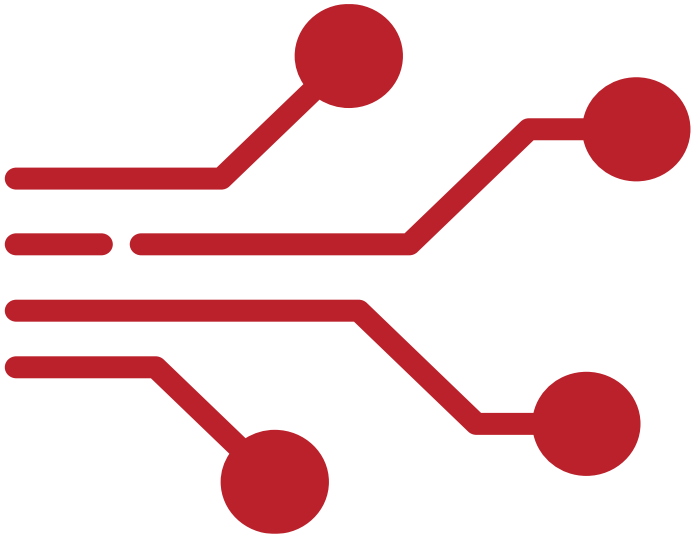
```
class Administrator : Uzytkownik
{
    public string Rola { get; set; } = "Admin";


    public void ZmienHaslo(string user)
    {
        Console.WriteLine($"{Login} zmienił hasło użytkownika:
{user}");
    }
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

329

Przykład

```
class SuperAdministrator : Administrator
{
    public void RestartujSerwer()
    {
        Console.WriteLine($"{Login} zrestartował serwer!");
    }
}
```



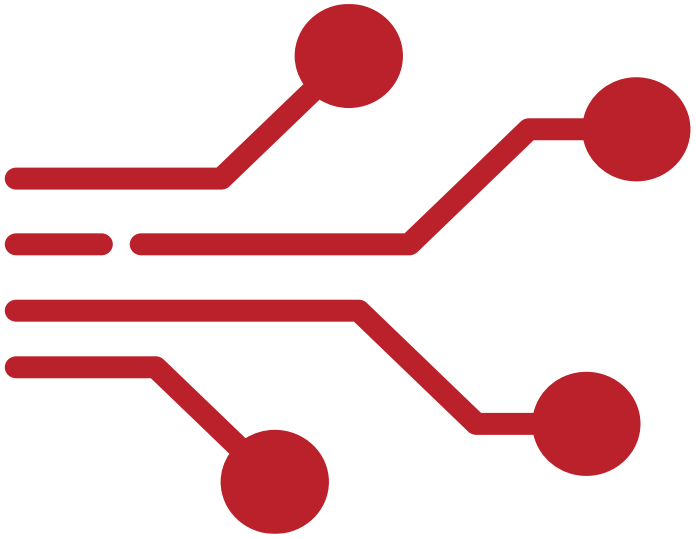
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

330

Przykład

```
class Program
{
    static void Main()
    {
        var u = new Uzytkownik { Login = "user01" };
        var admin = new Administrator { Login = "root" };
        var super = new SuperAdministrator { Login = "sysroot" };
        u.Zaloguj();
        admin.Zaloguj();
        admin.ZmienHaslo("user01");
        super.Zaloguj();
        super.RestartujSerwer();
    }
}
```

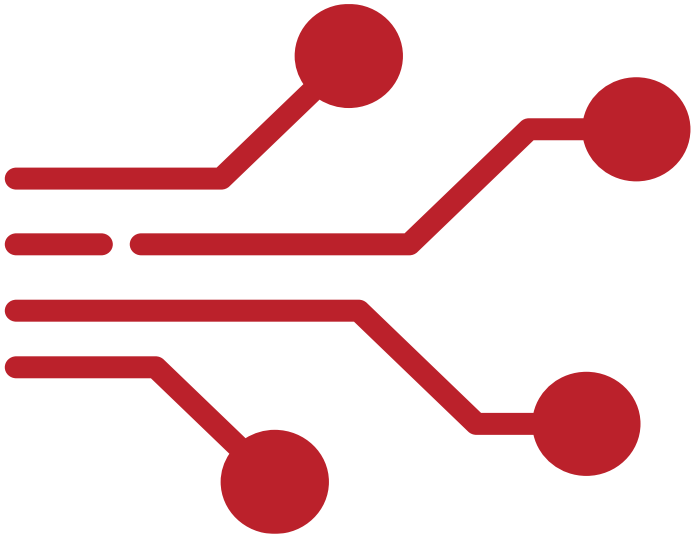


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

331

Przykład

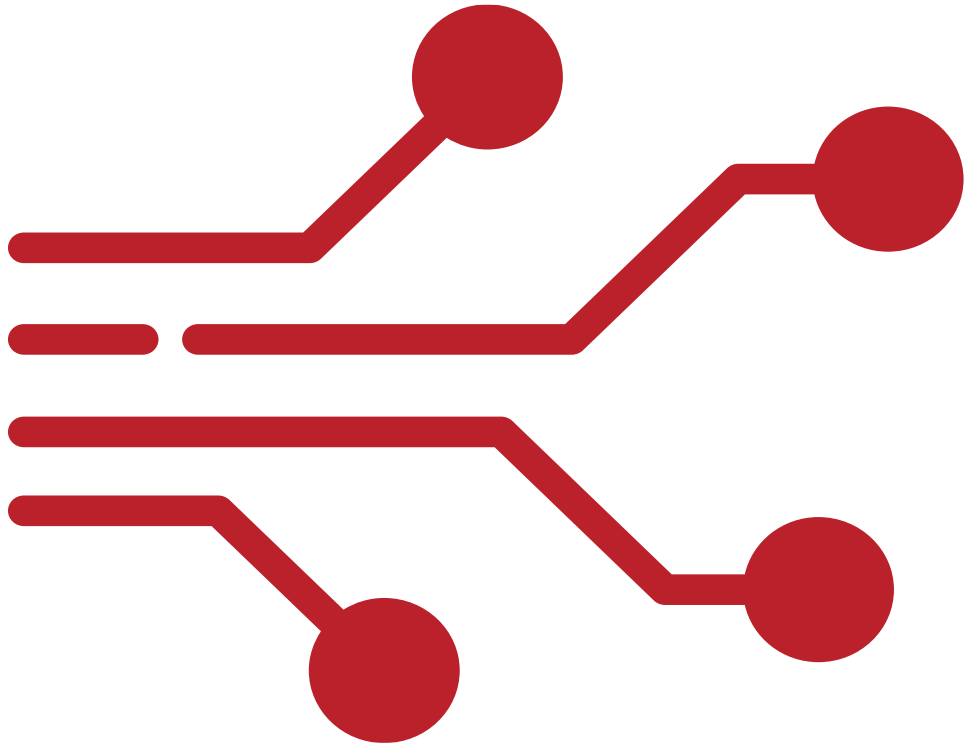
```
//Wynik

user01 zalogował się do systemu.
root zalogował się do systemu.
root zmienił hasło użytkownika: user01
sysroot zalogował się do systemu.
sysroot zrestartował serwer!
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

332

Polimorfizm



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

333

Słowa kluczowe virtual i override

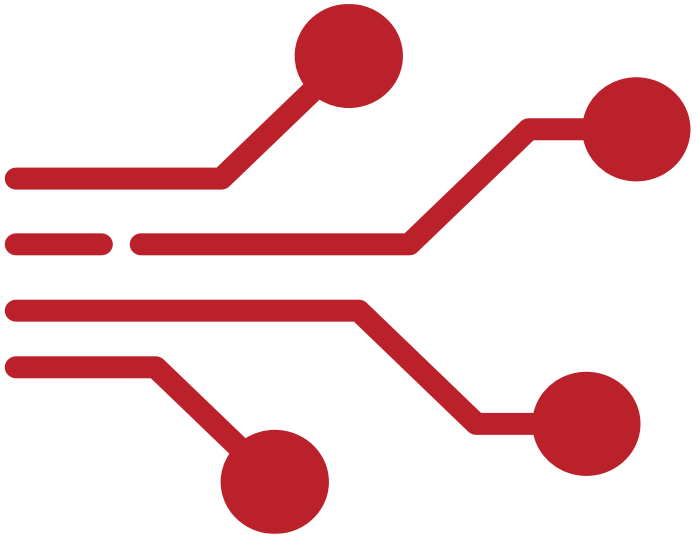
- virtual – metoda, którą można nadpisać w klasie pochodnej
- override – nadpisuje zachowanie metody wirtualnej z klasy bazowej
- Dzięki nim różne obiekty mogą reagować inaczej na to samo wywołanie
- Kluczowy mechanizm polimorfizmu (wiele form tego samego zachowania)
- Przykład: Zaloguj() u zwykłego użytkownika, administratora i audytora działa inaczej

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

334

Przykład

```
class Uzytkownik
{
    public string Login { get; set; }

    // metoda wirtualna
    public virtual void Zaloguj()
    {
        Console.WriteLine($"{Login} zalogował się standardowo.");
    }
}
```

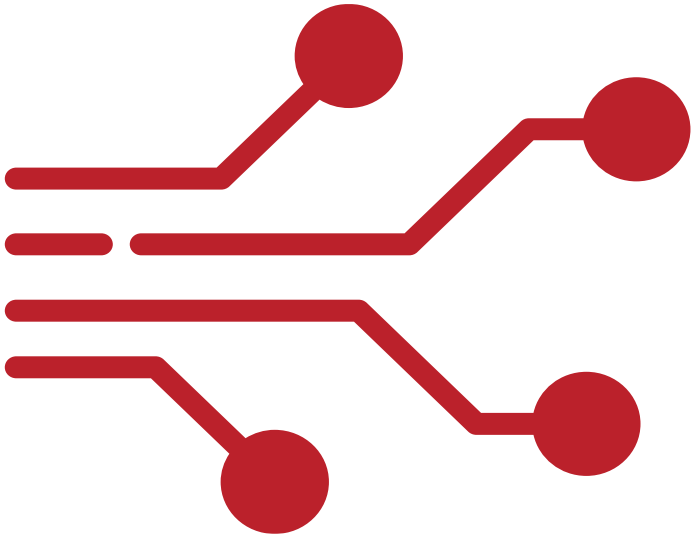


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

335

Przykład

```
class Administrator : Uzytkownik
{
    // metoda nadpisująca zachowanie
    public override void Zaloguj()
    {
        Console.WriteLine($"[ADMIN] {Login} zalogował się
            z uprawnieniami administratora.");
    }
}
```

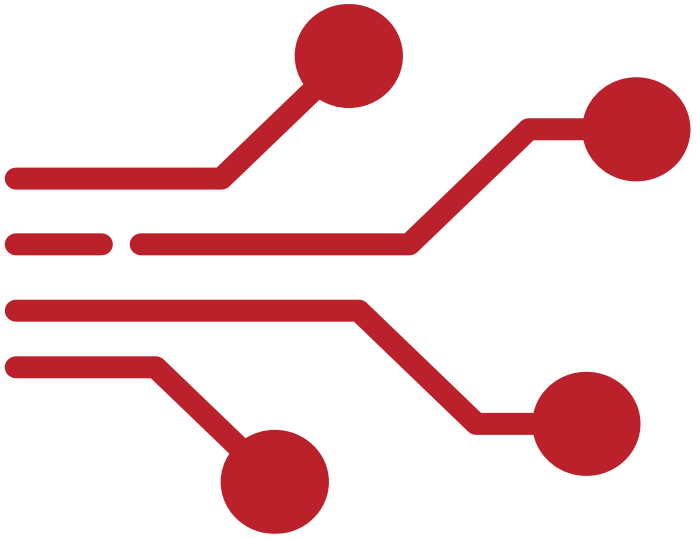


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

336

Przykład

```
class Audytor : Uzytkownik
{
    public override void Zaloguj()
    {
        Console.WriteLine($"[AUDYT] {Login} zalogował się w trybie
            monitorowania.");
    }
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

337

Przykład

```
class Program
{
    static void Main()
    {
        Uzytkownik u1 = new Uzytkownik { Login = "user01" };
        Uzytkownik u2 = new Administrator { Login = "root" };
        Uzytkownik u3 = new Audytor { Login = "audit" };

        u1.Zaloguj();
        u2.Zaloguj();
        u3.Zaloguj();
    }
}
```

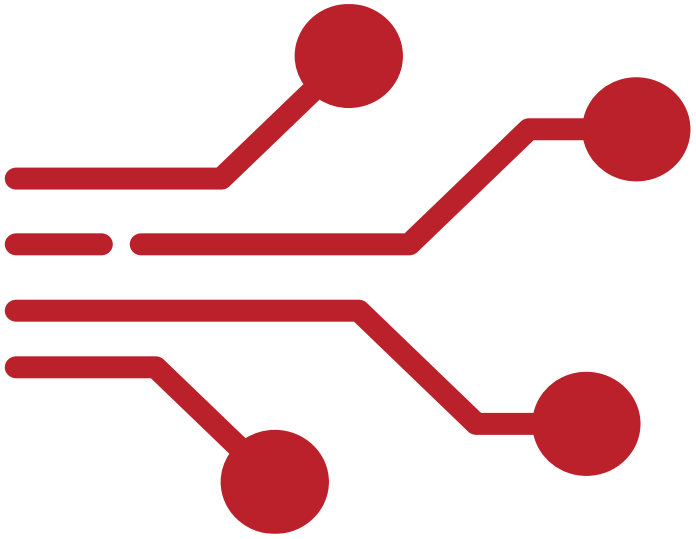

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

338

Przykład

```
//Wynik
user01 zalogował się standardowo.
[ADMIN] root zalogował się z uprawnieniami administratora.
[AUDYT] audit zalogował się w trybie monitorowania.
```



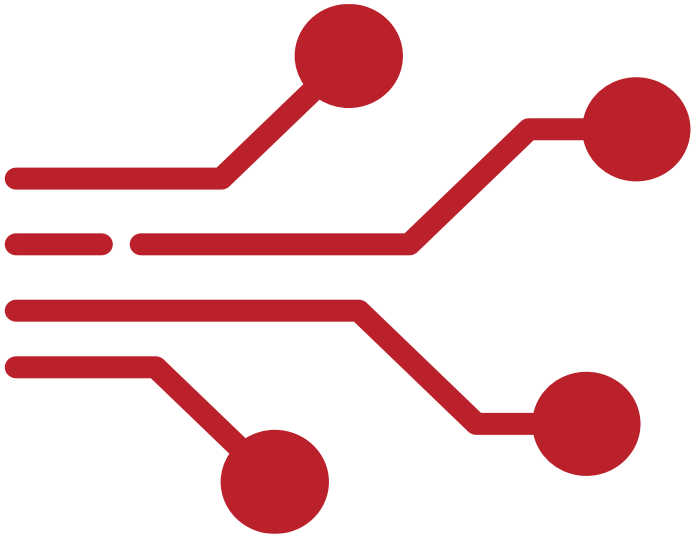
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

339

Polimorfizm – jedno wywołanie, wiele zachowań

- Polimorfizm to „wiele form” tego samego działania
- Obiekty różnych typów reagują inaczej na to samo wywołanie
- Możemy przechowywać różne typy obiektów w jednej kolekcji
- Dzięki virtual i override program sam wybiera odpowiednią metodę w czasie działania
- Kluczowa zaleta OOP: spójny kod, elastyczne zachowanie



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

340

Przykład

```
class Uzytkownik
{
    public string Login { get; set; }

    public virtual void Zaloguj()
    {
        Console.WriteLine($"{Login} zalogował się jako zwykły
            użytkownik.");
    }
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

341

Przykład

```
class Administrator : Uzytkownik{
    public override void Zaloguj()
    {
        Console.WriteLine($"[ADMIN] {Login} zalogował się z pełnymi
            uprawnieniami.");
    }
}
class Audytor : Uzytkownik{
    public override void Zaloguj()
    {
        Console.WriteLine($"[AUDYT] {Login} zalogował się tylko
            do odczytu logów systemowych.");
    }
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

342

Przykład

```
class Program
{
    static void Main()
    {
        // różne typy obiektów w jednej kolekcji
        List<Uzytkownik> konta = new()
        {
            new Uzytkownik { Login = "user01" },
            new Administrator { Login = "root" },
            new Audytor { Login = "sec_aud" }
        };
```

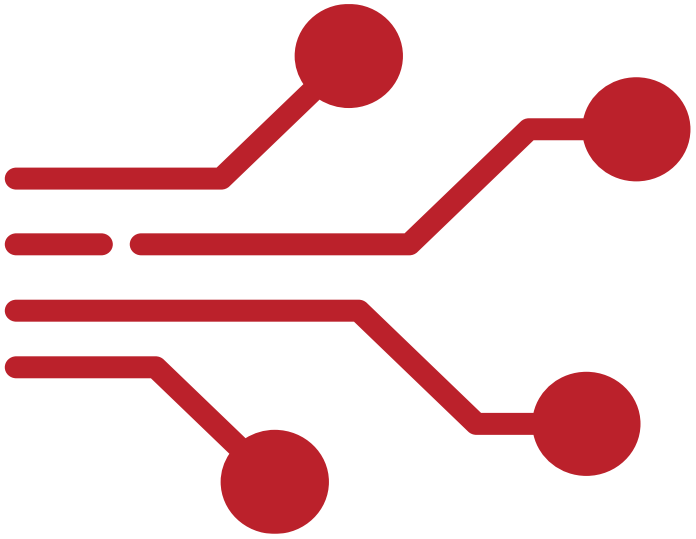

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

343

Przykład

```
            foreach (var konto in konta)
            {
                konto.Zaloguj(); // jedno wywołanie, różne zachowania
            }
        }
}
```

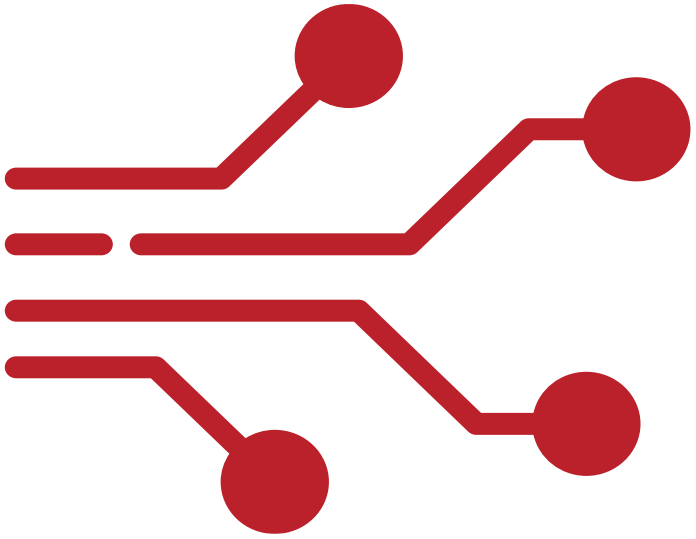


```
//Wynik
user01 zalogował się jako zwykły użytkownik.
[ADMIN] root zalogował się z pełnymi uprawnieniami.
[AUDYT] sec_aud zalogował się tylko do odczytu logów systemowych.
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

344



Klasy abstrakcyjne (abstract)

- Klasa abstrakcyjna to klasa, dla której nie można utworzyć instancji
- Zawiera wspólne pola, właściwości i metody (czasem bez implementacji)
- Może definiować metody abstrakcyjne – bez ciała
- Klasy pochodne muszą te metody zaimplementować (override)
- Służy jako szkielet dla grupy powiązanych klas
- Idealna, gdy chcemy wymusić spójność zachowań między różnymi typami

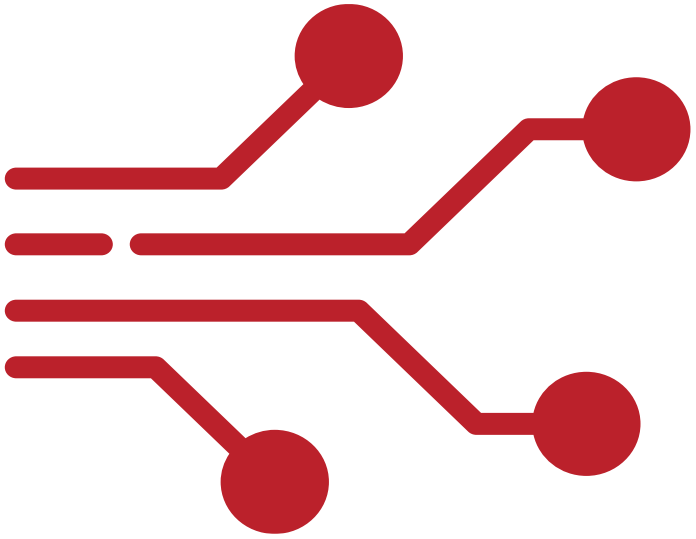
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

345

Metody abstrakcyjne i ich implementacja

- Metoda abstrakcyjna – bez implementacji w klasie bazowej
- Klasy pochodne muszą ją zaimplementować (override)
- Nie można użyć metody abstrakcyjnej bezpośrednio
- Pozwalają wymusić spójne zachowanie w całej hierarchii
- Idealne do systemów bezpieczeństwa – np. każda klasa musi implementować analizę zdarzenia lub logowanie incydentu

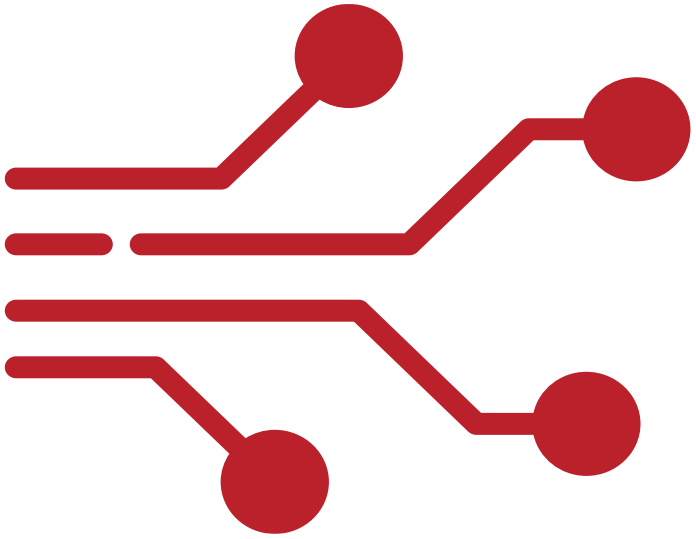


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

346

Interfejsy i ich rola w OOP



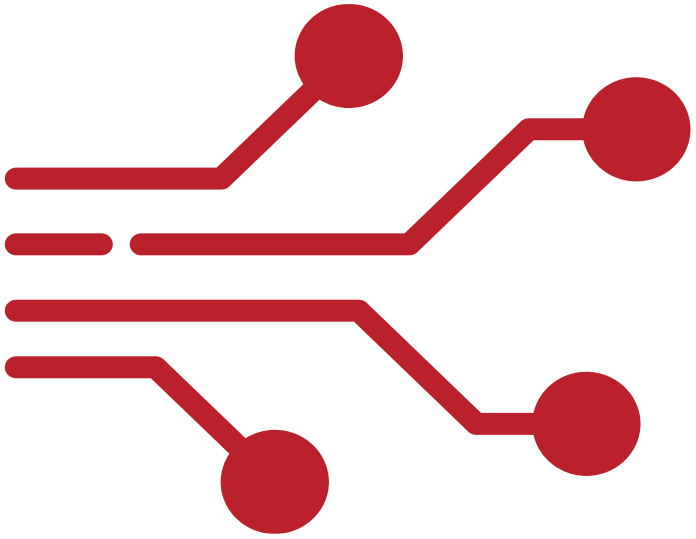
- Interfejs (interface) – zbiór deklaracji metod bez implementacji
- Klasa, która implementuje interfejs, musi zaimplementować wszystkie jego metody
- Interfejs nie zawiera pól ani konstruktorów
- Klasa może implementować wiele interfejsów (brak ograniczenia jak w dziedziczeniu klas)
- Służy do definiowania kontraktów – co obiekt musi umieć robić

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

347

Dziedziczenie wielopoziomowe i interfejsy w jednym projekcie



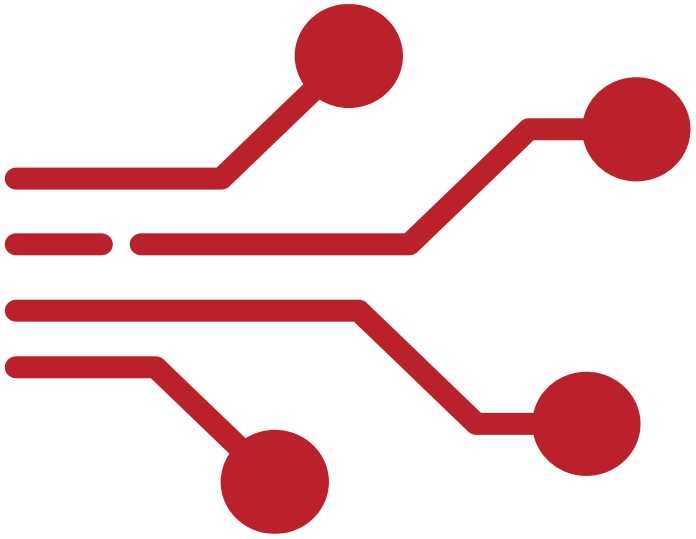
- Klasy mogą jednocześnie dziedziczyć po innej klasie i implementować wiele interfejsów
- Pozwala to łączyć strukturę (dziedziczenie) z funkcjonalnością (interfejsy)
- Klasa bazowa definiuje cechy wspólne
- Interfejsy definiują zachowania wymagane przez system
- W praktyce: Administrator dziedziczy po Uzytkownik i implementuje np. IAutoryzowalny, IAudytowalny

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

348

Kiedy używać dziedziczenia?

- Kiedy istnieje logiczny związek „jest”
- Kiedy chcemy współdzielić zachowanie
- Kiedy klasy potomne mają podobną strukturę
- **Unikać**, jeśli klasy tylko przypadkowo mają podobne metody



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

349

Dziękuję za uwagę

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Wykład nr 9
Temat: Zdarzenia i wyjątki

Paweł Karczmarek
Katedra Inteligencji Obliczeniowej Politechniki Lubelskiej

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Plan prezentacji

- Zdarzenia
- Wyjątki

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

352

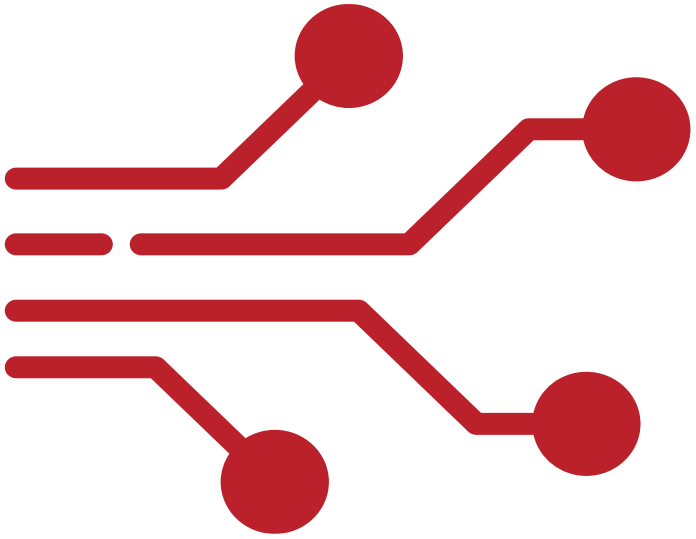
Zdarzenia

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

353

Wprowadzenie do zdarzeń



- Zdarzenie (event) to sygnał, że w systemie zaszło jakieś działanie
- Program może nasłuchiwać zdarzeń i reagować na nie
- Zdarzenia uruchamiają funkcje (reakcje) – np. alarm po błędnym logowaniu
- W C# zdarzenia są oparte na delegatach (czyli wskaźnikach na metody)
- Model: nadawca (publisher) -> odbiorca (subscriber)

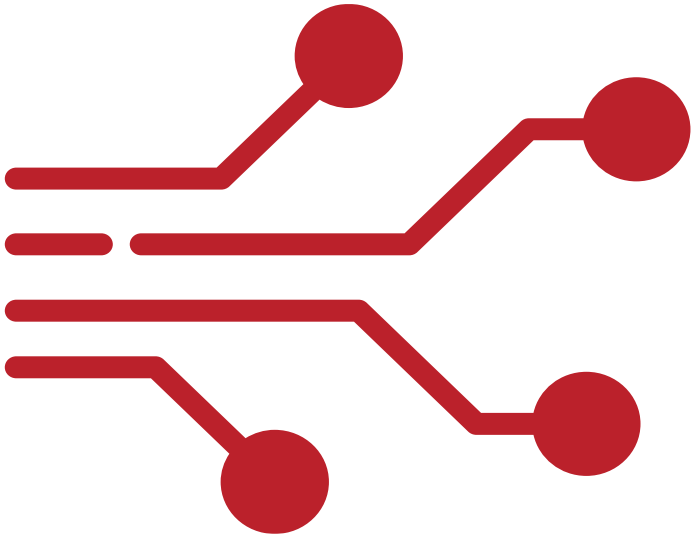
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

354

Model zdarzeniowy w C#: Publisher i Subscriber

- Publisher (nadawca) – obiekt, który wywołuje zdarzenie
- Subscriber (odbiorca) – obiekt, który nasłuchuje zdarzenia
- Gdy zdarzenie wystąpi, C# automatycznie wywołuje powiązaną metodę
- Mechanizm oparty na delegatach i eventach
- Typowy układ: źródło -> zdarzenie -> reakcja



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

355

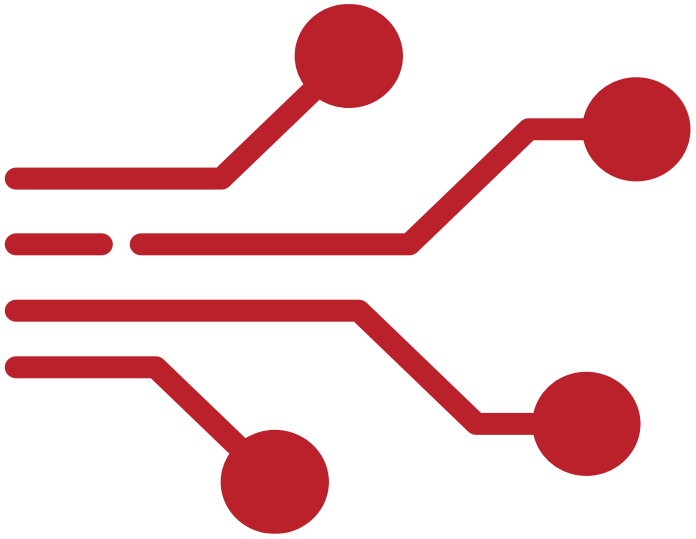
Przykład

```
using System;
class Alarm
{
    // Definicja zdarzenia
    public event Action OnIntrusionDetected;
    // Wywołanie zdarzenia
    public void WykryjIntruz()
    {
        Console.WriteLine("Wykryto ruch w strefie
            chronionej!");
        OnIntrusionDetected?.Invoke();
    }
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Przykład

```
class Monitor
{
    // Reakcja na zdarzenie
    public void Reaguj()
    {
        Console.WriteLine("Monitor: Uruchamiam alarm
            i zapisuję zdarzenie do logu!");
    }
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

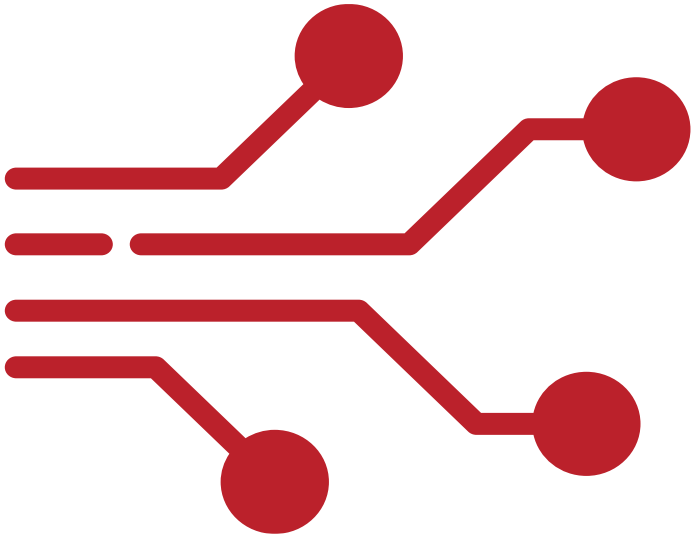
357

Przykład

```
class Program
{
    static void Main()
    {
        Alarm alarm = new Alarm();
        Monitor monitor = new Monitor();
        // Subskrypcja – monitor nasłuchuje zdarzenia
        alarm.OnIntrusionDetected += monitor.Reaguj;
        // Symulacja zdarzenia
        alarm.WykryjIntruz();
    }
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Przykład

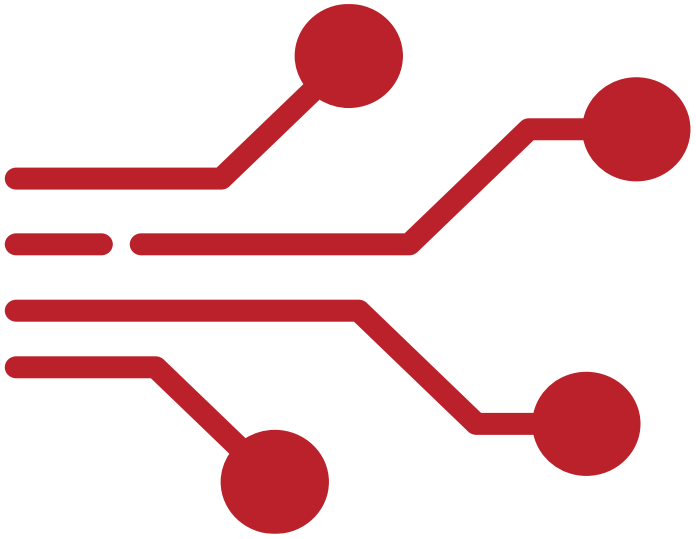


```
// Wynik

Wykryto ruch w strefie chronionej!
Monitor: Uruchamiam alarm i zapisuję zdarzenie do logu!
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

359



Delegaty a zdarzenia

- Delegat to typ, który może przechowywać odwołanie do metody
- Dzięki delegatom możemy wywoływać metody pośrednio
- Zdarzenia używają delegatów do uruchamiania reakcji
- delegate + event = dynamiczny system reakcji na sytuacje w programie
- W cyberbezpieczeństwie: delegaty to zdefiniowane „reakcje” systemu

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

360

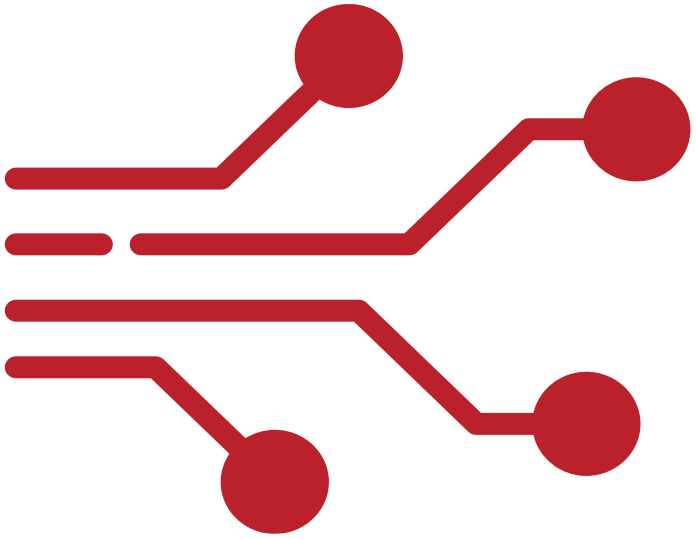
Przykład – najprostszy delegat

```
using System;
delegate void ReakcjaNaAtak(); // definicja typu delegata
class SystemBezpieczenstwa
{
    public ReakcjaNaAtak reakcja; // pole typu delegata
    public void WykrytoAtak()
    {
        Console.WriteLine("Atak wykryty!");
        reakcja?.Invoke(); // wywołanie metody przypisanej
            //do delegata
    }
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Przykład – najprostszy delegat

```
class Program
{
    static void Alarm()
    {
        Console.WriteLine("Alarm włączony!");
    }
    static void PowiadomAdministratora()
    {
        Console.WriteLine("Wysłano powiadomienie
            do administratora.");
    }
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

362

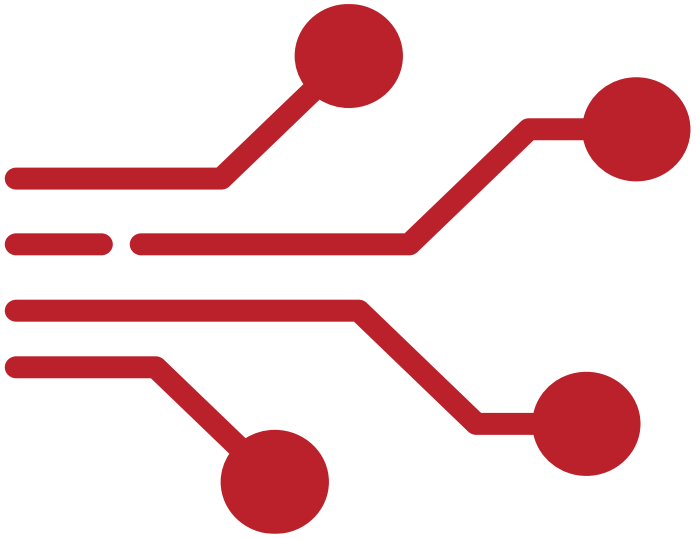
Przykład – najprostszy delegat

```
        static void Main()
        {
            var system = new SystemBezpieczenstwa();

            // przypisanie metody do delegata
            system.reakcja = Alarm;
            system.reakcja += PowiadomAdministratora;

            system.WykrytoAtak();
        }
    }
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Przykład – najprostszy delegat

```
//Wynik w konsoli
Atak wykryty!
Alarm włączony!
Wysłano powiadomienie do administratora.
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

364

Definicja i wywołanie zdarzenia

- event to mechanizm sygnalizacji w C# – informuje o tym, że coś się stało
- Wymaga delegata (np. Action, EventHandler)
- Inne obiekty mogą subskrybować zdarzenie (operator +=)
- Zdarzenie można wywołać tylko wewnątrz klasy, w której zostało zdefiniowane
- Typowe wzorce: OnSomethingHappened() i EventArgs

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

365

Przykład – prosty system alarmowy

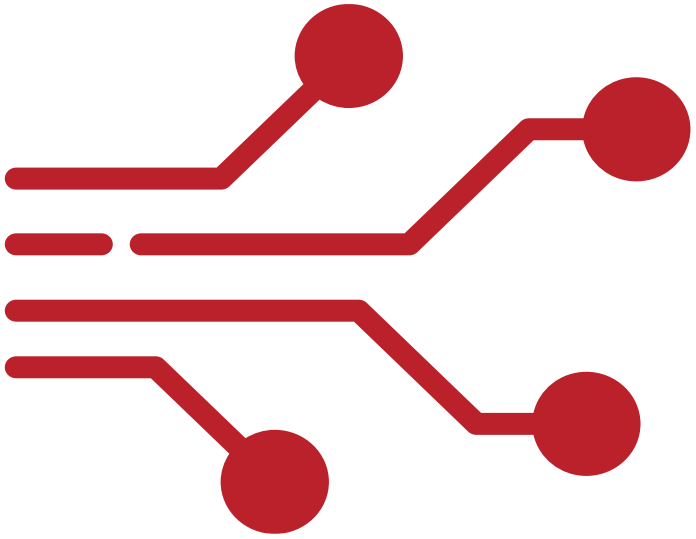
```
using System;
class Alarm
{
    // Definicja zdarzenia
    public event Action OnIntrusionDetected;
    // Metoda wywołująca zdarzenie
    public void SprawdzSystem()
    {
        Console.WriteLine("Sprawdzanie czujników...");
        bool wykrytoRuch = true;
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

366

Przykład – prosty system alarmowy

```
            if (wykrytoRuch)
            {
                Console.WriteLine("Wykryto ruch! Uruchamiam
                 zdarzenie.");
                OnIntrusionDetected?.Invoke(); // wywołanie
                 //eventu
            }
        }
    }
```

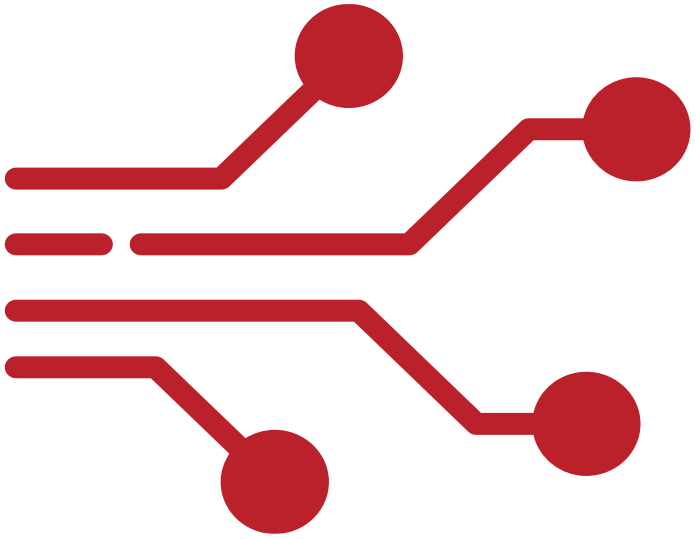


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

367

Przykład – prosty system alarmowy

```
class ReakcjaBezpieczenstwa
{
    // Metoda reagująca na zdarzenie
    public void UruchomAlarm()
    {
        Console.WriteLine("Reakcja: Uruchamiam alarm
            i wysyłam powiadomienie!");
    }
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

368

Przykład – prosty system alarmowy

```
class Program
{
    static void Main()
    {
        Alarm alarm = new Alarm();
        ReakcjaBezpieczenstwa reakcja = new
            ReakcjaBezpieczenstwa();
        // Subskrypcja zdarzenia
        alarm.OnIntrusionDetected += reakcja.UruchomAlarm;
        // Symulacja działania
        alarm.SprawdzSystem();
    }
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

369

Subskrypcja i reakcja na zdarzenie

```
//Wynik w konsoli
Sprawdzanie czujników...
Wykryto ruch! Uruchamiam zdarzenie.
Reakcja: Uruchamiam alarm i wysyłam powiadomienie!
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

370

Subskrypcja i reakcja na zdarzenie

- Zdarzenie może mieć wielu subskrybentów
- Każdy subskrybent wykonuje własną reakcję
- Subskrypcja: += oznacza dołącz metodę
- Rezygnacja: -= oznacza odłącz metodę
- W C# event wywołuje wszystkie zapisane reakcje – kolejno, bez zależności między nimi

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

371

Przykład – wiele reakcji na jedno zdarzenie

```
using System;
class Alarm
{
    public event Action OnIntrusionDetected;
    public void WykryjIntruz()
    {
        Console.WriteLine("Wykryto próbę włamania!");
        OnIntrusionDetected?.Invoke();
    }
}
```

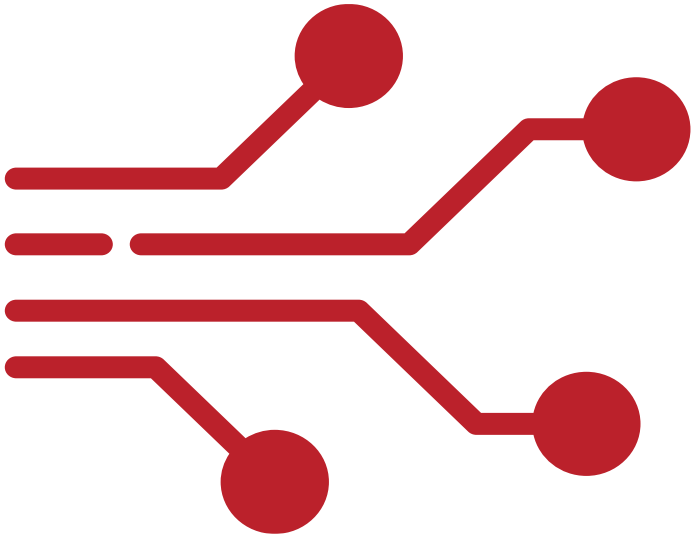

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

372

Przykład – wiele reakcji na jedno zdarzenie

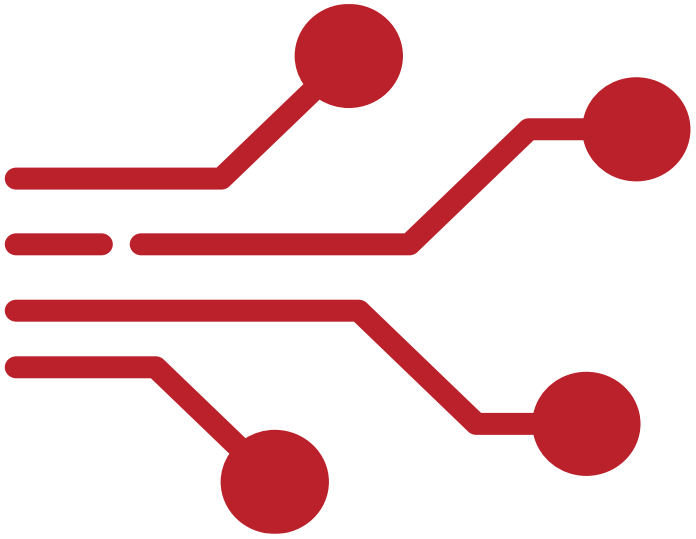
```
class ReakcjaSyrena
{
    public void Uruchom()
    {
        Console.WriteLine("Syrena: Włączam alarm
            dźwiękowy!");
    }
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

373

Przykład – wiele reakcji na jedno zdarzenie



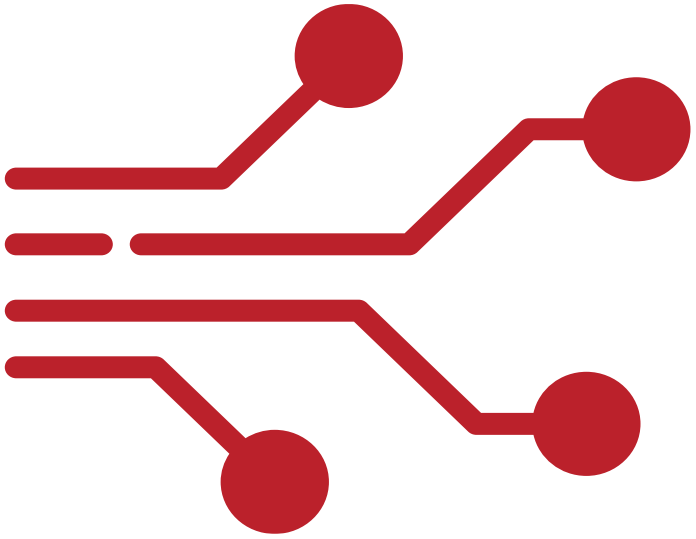
```
class ReakcjaPowiadomienie
{
    public void WyslijEmail()
    {
        Console.WriteLine("Powiadomienie: Wysłano e-mail
                           do administratora.");
    }
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

374

Przykład – wiele reakcji na jedno zdarzenie

```
class ReakcjaLog
{
    public void ZapiszLog()
    {
        Console.WriteLine("Log: Zdarzenie zapisane
                           w dzienniku bezpieczeństwa.");
    }
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

375

Przykład – wiele reakcji na jedno zdarzenie

```
class Program
{
    static void Main()
    {
        Alarm alarm = new Alarm();
        var syrena = new ReakcjaSyrena();
        var mail = new ReakcjaPowiadomienie();
        var log = new ReakcjaLog();
        // Subskrypcje – wiele reakcji na jedno zdarzenie
        alarm.OnIntrusionDetected += syrena.Uruchom;
        alarm.OnIntrusionDetected += mail.WyslijEmail;
        alarm.OnIntrusionDetected += log.ZapiszLog;
```

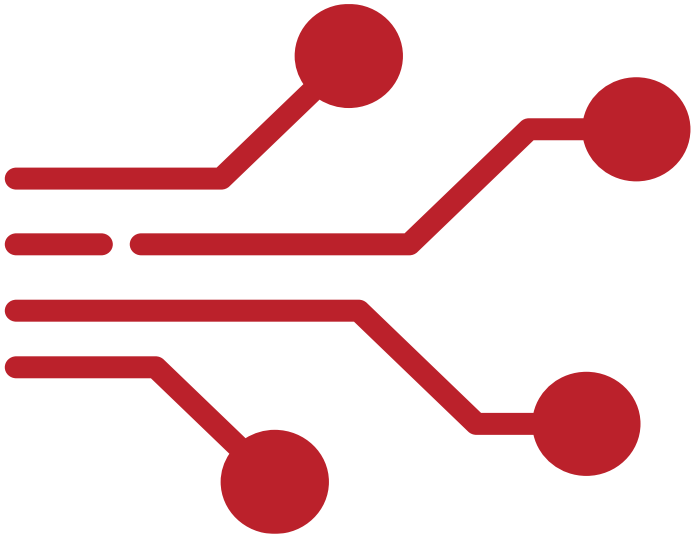

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

376

Przykład – wiele reakcji na jedno zdarzenie

```
        // Symulacja incydentu
        alarm.WykryjIntruz();
    }
}
```

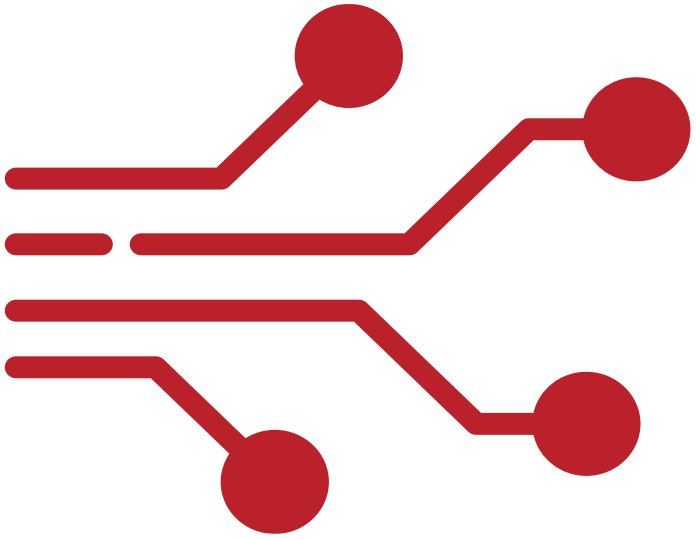


```
//Wynik
Wykryto próbę włamania!
Syrena: Włączam alarm dźwiękowy!
Powiadomienie: Wysłano e-mail do administratora.
Log: Zdarzenie zapisane w dzienniku bezpieczeństwa.
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Własne zdarzenia z argumentami (EventArgs)

- Zdarzenie może przekazywać dodatkowe informacje
- W tym celu używamy klasy EventArgs lub jej pochodnych
- Argumenty pozwalają określić kto, co i kiedy wywołał zdarzenie
- Dzięki temu reakcje mogą być lepiej określone
- W C#: event EventHandler<CustomEventArgs>



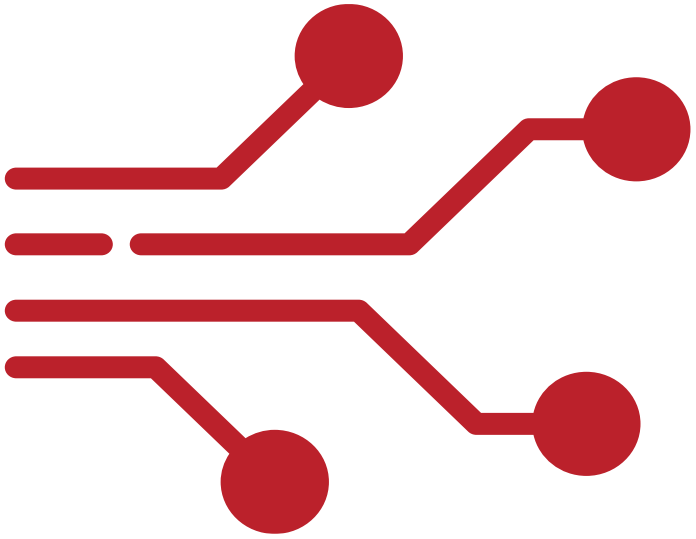
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

378

Przykład – wykrycie nieudanej próby logowania

```
using System;
// Klasa argumentów zdarzenia
class LoginEventArgs : EventArgs
{
    public string Uzytkownik { get; }
    public string AdresIP { get; }
    public DateTime Czas { get; }
    public LoginEventArgs(string uzytkownik, string adresIP)
    {
        Uzytkownik = uzytkownik;
        AdresIP = adresIP;
        Czas = DateTime.Now;
    }
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

379

Przykład – wykrycie nieudanej próby logowania

```
// Klasa generująca zdarzenie
class SystemLogowania
{
    public event EventHandler<LoginEventArgs> OnLoginFailed;
    public void Loguj(string user, string pass, string ip)
    {
        Console.WriteLine($"Próba logowania użytkownika: {user}");
        if (pass != "qwerty456")
        {
            Console.WriteLine("Niepoprawne hasło!");
            OnLoginFailed?.Invoke(this, new LoginEventArgs(user,
               ip));
        }
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Przykład – wykrycie nieudanej próby logowania

```
            else
            {
                Console.WriteLine("Zalogowano pomyślnie.");
            }
        }
}
// Klasa reagująca na zdarzenie
class SystemBezpieczenstwa
{
    public void AnalizujBłąd(object sender, LoginEventArgs e){
        Console.WriteLine($"ALERT: Nieudane logowanie użytkownika
{e.Uzytkownik} z adresu {e.AdresIP} o {e.Czas}!");
    }
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

381

Przykład – wykrycie nieudanej próby logowania

```
// Program główny
class Program
{
    static void Main()
    {
        var system = new SystemLogowania();
        var bezpieczenstwo = new SystemBezpieczenstwa();
        // Subskrypcja
        system.OnLoginFailed += bezpieczenstwo.AnalizujBłąd;
        // Symulacja logowania
        system.Loguj("jan", "qwerty", "192.168.0.5");
    }
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

382

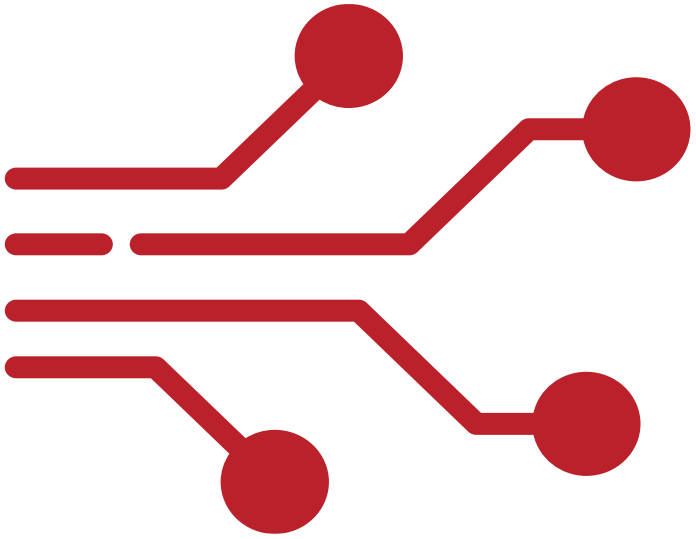
Przykład – wykrycie nieudanej próby logowania

```
//Wynik
Próba logowania użytkownika: jan
Niepoprawne hasło!
ALERT: Nieudane logowanie użytkownika jan z adresu 192.168.0.5
o 07.10.2025 16:42:11!
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

383

Zdarzenia systemowe – monitorowanie plików (FileSystemWatcher)



- FileSystemWatcher to klasa z przestrzeni nazw System.IO
- Umożliwia reagowanie na zmiany w systemie plików: tworzenie, modyfikację, usuwanie, zmianę nazwy
- Działa asynchronicznie – nie blokuje programu
- Wysyła zdarzenia: Changed, Created, Deleted, Renamed
- Może monitorować określony folder lub cały dysk

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

384

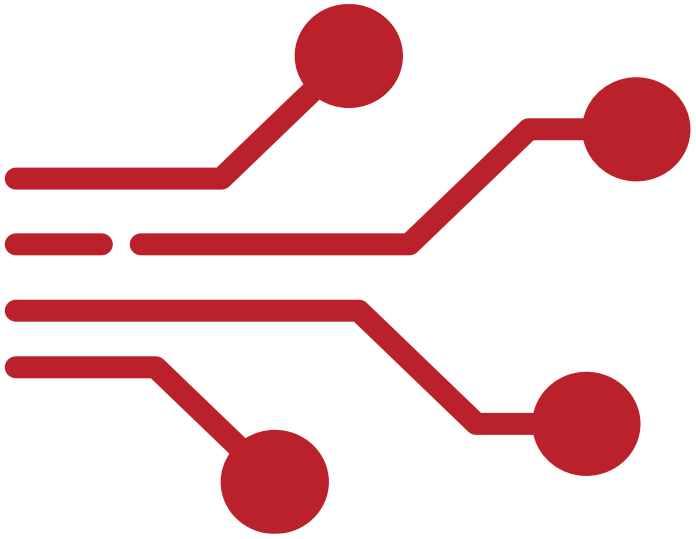
Przykład

```
using System;
using System.IO;
class Program
{
    static void Main()
    {
        // Tworzymy obserwatora
        FileSystemWatcher watcher = new FileSystemWatcher();
        // Określamy folder i typ plików do monitorowania
        watcher.Path = "."; // bieżący katalog
        watcher.Filter = "security.log"; // tylko ten plik
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Przykład



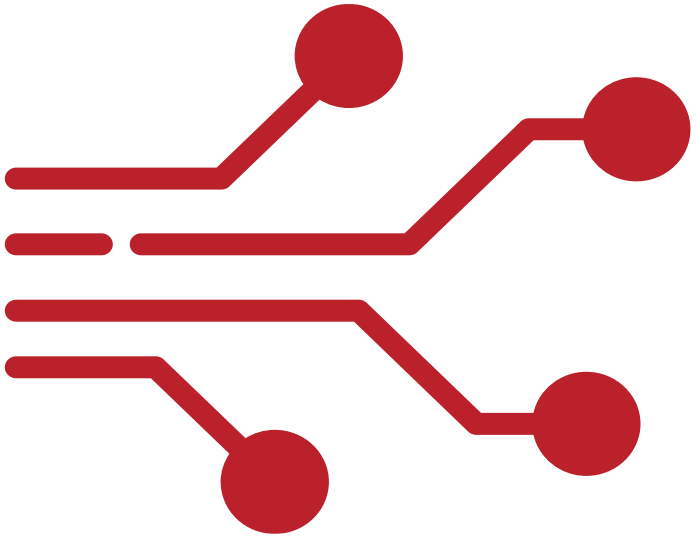
```
watcher.IncludeSubdirectories = false;
     // nie monitoruj podfolderów
// Wybieramy, jakie zmiany nas interesują
watcher.NotifyFilter = NotifyFilters.LastWrite |
     // zmiana zawartości
     NotifyFilters.FileName |    // zmiana nazwy
     NotifyFilters.Size;         // zmiana rozmiaru
// Subskrybujemy zdarzenia
watcher.Changed += OnChanged;
watcher.Created += OnCreated;
watcher.Deleted += OnDeleted;
watcher.Renamed += OnRenamed;
watcher.Error += OnError;
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

386

Przykład

```
        // Uruchamiamy obserwację
        watcher.EnableRaisingEvents = true;
        Console.WriteLine("System monitoruje plik
            security.log");
        Console.WriteLine("Wciśnij Enter, aby zakończyć...");
        Console.ReadLine();
    }
```



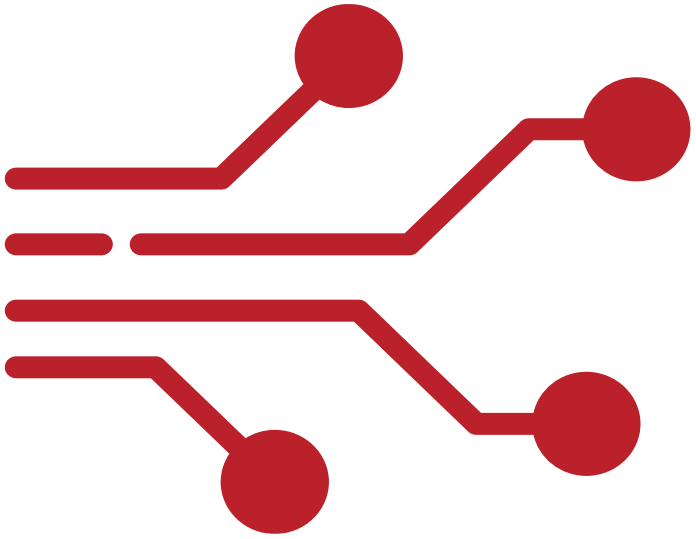
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

387

Przykład

```
// Reakcje na zdarzenia

   private static void OnChanged(object sender,
      FileSystemEventArgs e)

   {

       Console.WriteLine($"[CHANGED] {e.FullPath} –
            zawartość zmodyfikowana o {DateTime.Now:T}");

   }

   private static void OnCreated(object sender,
      FileSystemEventArgs e)

   {

       Console.WriteLine($"[CREATED] {e.FullPath} – plik
            utworzony.");

   }
```



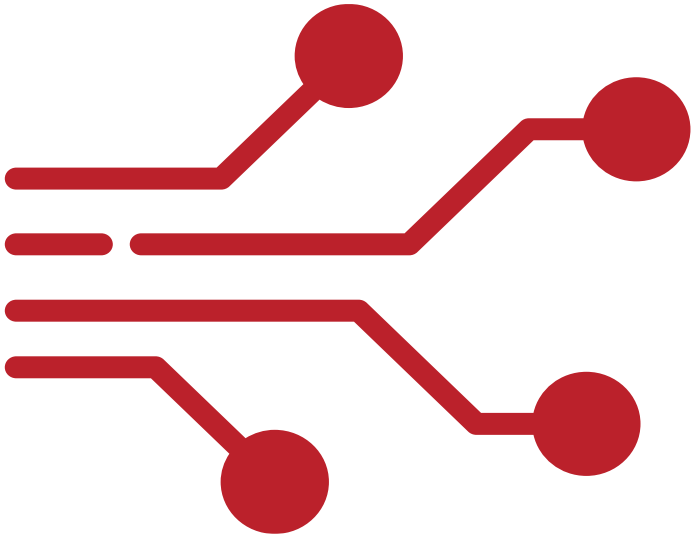
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

388

Przykład

```
private static void OnDeleted(object sender,
   FileSystemEventArgs e)
{
    Console.WriteLine($"[DELETED] {e.FullPath} – plik
        usunięty.");
}
private static void OnRenamed(object sender,
   RenamedEventArgs e)
{
    Console.WriteLine($"[RENAMED] {e.OldName} ->
        {e.Name}");
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

389

Przykład

```
    private static void OnError(object sender, ErrorEventArgs
       e)
    {
        Console.WriteLine($"[ERROR] Błąd obserwatora:
            {e.GetException().Message}");
    }
}

//Tworzenie obserwatora
FileSystemWatcher watcher = new FileSystemWatcher();
```

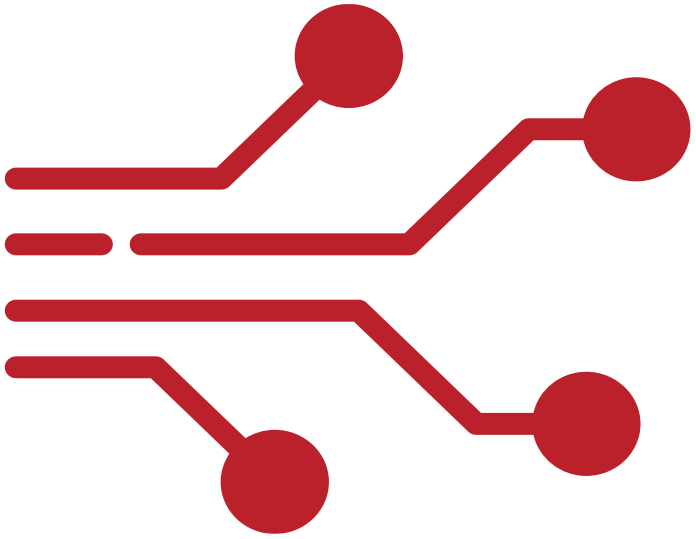

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

390

Timer: zdarzenia czasowe w aplikacji konsolowej

- Timer generuje zdarzenia czasowe (Elapsed) w określonych odstępach
- Każde „tyknięcie” to event – system wywołuje naszą metodę automatycznie
- Nie używamy pętli ani opóźnień, tylko subskrypcję zdarzenia
- Timer działa asynchronicznie – nie blokuje programu
- Jest idealny do cyklicznego sprawdzania logów, statusu, alertów



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

391

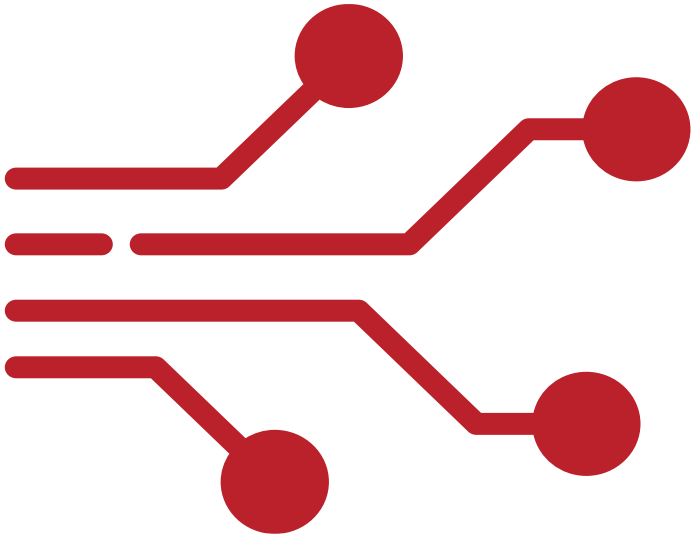
Przykład

```
using System;
using System.Timers; // przestrzeń nazw z klasą Timer
class Program
{
    static void Main()
    {
        // Tworzymy timer co 3 sekundy
        Timer timer = new Timer(3000);
        // Subskrybujemy zdarzenie Elapsed (czyli "upłynął
           czas")
        timer.Elapsed += OnTick;
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

392

Przykład



```
        // Uruchamiamy timer
        timer.AutoReset = true;   // powtarzaj cyklicznie
        timer.Enabled = true;     // aktywuj
        Console.WriteLine("Timer uruchomiony. Wciśnij Enter,
             aby zakończyć.");
        Console.ReadLine();
             // program działa, dopóki nie naciśniemy Enter
        // Zatrzymanie i czyszczenie
        timer.Stop();
        timer.Dispose();
    }
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

393

Przykład

```
    // Reakcja na zdarzenie
    private static void OnTick(object? sender,
       ElapsedEventArgs e)
    {
        Console.WriteLine($"[{e.SignalTime:HH:mm:ss}]
            Wykryto zdarzenie czasowe!");
    }
}
//Wynik
Timer uruchomiony. Wciśnij Enter, aby zakończyć.
[16:41:13] Wykryto zdarzenie czasowe!
[16:41:16] Wykryto zdarzenie czasowe!
[16:41:19] Wykryto zdarzenie czasowe!
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

394

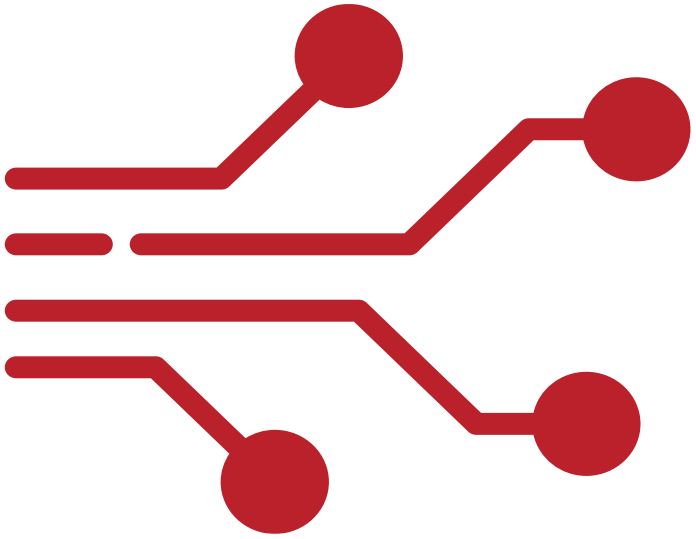
Wyjątki

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Co to jest wyjątek? Różnica między błędem a sytuacją wyjątkową

- Błąd (error) to sytuacja niepoprawna, której program nie potrafi kontynuować
- Wyjątek (exception) to kontrolowany sposób powiadomienia o błędzie
- Wyjątek nie kończy programu natychmiast — daje szansę na reakcję
- C# obsługuje wyjątki za pomocą try, catch, finally, throw
- Wyjątki to „zdarzenia awaryjne” – system reaguje, zamiast się zawieszać



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

396

Przykład (bez obsługi/z obsługą wyjątku)

```
int a = 10;
int b = 0;
int wynik = a / b; // DivideByZeroException
Console.WriteLine("Wynik: " + wynik);
try
{
    int a = 10, b = 0;
    int wynik = a / b;
    Console.WriteLine("Wynik: " + wynik);
}
catch (DivideByZeroException)
{
    Console.WriteLine("Błąd: Nie można dzielić przez zero!");
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

397

Mechanizm try-catch-finally

- try – blok kodu, w którym może wystąpić wyjątek
- catch – blok, który obsługuje błąd
- finally – blok, który zawsze się wykona (np. sprzątanie, zamknięcie połączenia)
- C# umożliwia kilka catch dla różnych typów błędów
- finally działa zawsze — nawet po błędzie lub return

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

398

Przykład

```
int a = 10;
int b = 0;
try
{
    Console.WriteLine("Próbuję wykonać dzielenie...");
    int wynik = a / b;
    Console.WriteLine("Wynik: " + wynik);
}
catch (DivideByZeroException)
{
    Console.WriteLine("Błąd: Próba dzielenia przez zero!");
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

399

Przykład

```
finally
{
    Console.WriteLine("Zakończono operację (sprzątanie, zwolnienie
        zasobów).");
}


//Wynik
Próbuję wykonać dzielenie...
Błąd: Próba dzielenia przez zero!
Zakończono operację (sprzątanie, zwolnienie zasobów).
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

400

Przykład

```
using System;
using System.IO;
class Program
{
    static void Main()
    {
        try
        {
            Console.WriteLine("Próba odczytu pliku...");
            string dane = File.ReadAllText("brakujący_plik.txt");
            Console.WriteLine("Próba dzielenia...");
            int wynik = 10 / int.Parse("0");
        }
```

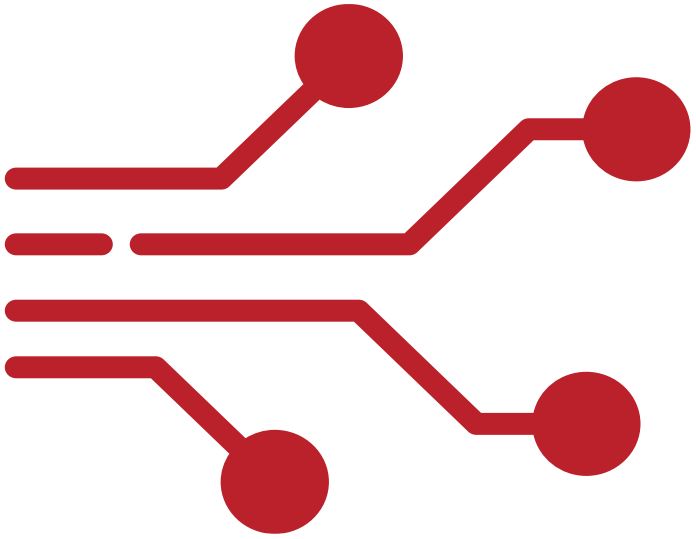

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

401

Przykład

```
catch (FileNotFoundException ex)
{
    Console.WriteLine($"Błąd pliku: {ex.Message}");
}
catch (DivideByZeroException)
{
    Console.WriteLine("Błąd: Nie można dzielić przez 0!");
}
catch (FormatException)
{
    Console.WriteLine("Błąd formatu danych!");
}
```



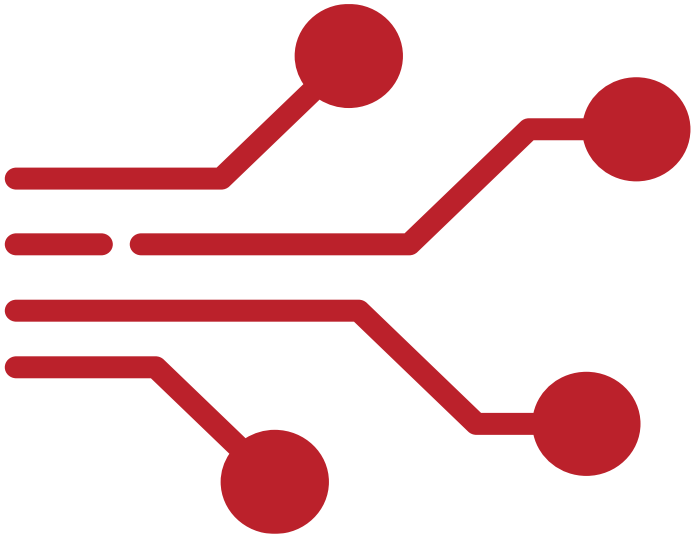
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

402

Przykład

```
        catch (Exception ex)
        {
            Console.WriteLine($"Inny błąd: {ex.GetType().Name}");
        }
        finally
        {
            Console.WriteLine("🧹 Zakończono obsługę błędów.");
        }
    }
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

403

Słowo kluczowe throw i propagacja wyjątków

- throw służy do zgłaszania wyjątku
- Wyjątek może być rzucony w jednym miejscu, a obsłużony w innym
- Nieobsłużony wyjątek propaguje się w górę stosu wywołań
- throw; ponownie zgłasza ten sam wyjątek (zachowuje kontekst)
- throw ex; zgłasza wyjątek ponownie, ale traci ślad (stack trace)

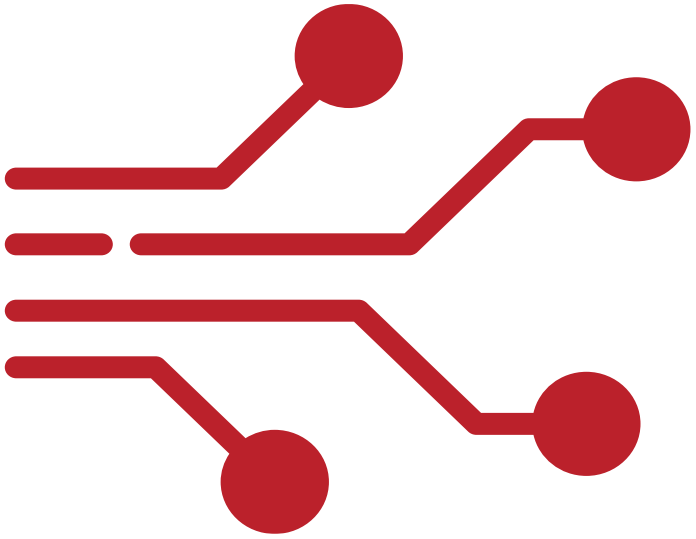
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

404

Przykład

```
using System;
class Program
{
    static void Main()
    {
        try
        {
            MetodaA();
        }
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

405

Przykład

```
        catch (Exception ex)
        {
            Console.WriteLine($"Główny blok: przechwycono
             wyjątek -> {ex.GetType().Name}");
        }
    }
    static void MetodaA()
    {
        Console.WriteLine("Wywołano MetodaA()");
        MetodaB();
    }
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

406

Przykład

```
    static void MetodaB()
    {
        Console.WriteLine("Wywołano MetodaB()");
        throw new InvalidOperationException("Nieprawidłowa
            operacja!");
    }
}
//Wynik
Wywołano MetodaA()
Wywołano MetodaB()
Główny blok: przechwycono wyjątek->InvalidOperationException
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Dobre praktyki obsługi wyjątków

- Nie powinno się ignorować wyjątków, ale zawsze je obsługiwać lub logować
- Nie powinno się ujawniać szczegółów błędu użytkownikowi (ścieżki, SQL, dane)
- Zawsze należy logować szczegóły błędu do systemu audytu/pliku logu/tabeli w bazie
- Nie używa się catch (Exception) bez powodu
- Należy używać finally do zwalniania zasobów
- Należy tworzyć własne typy wyjątków dla specyficznych sytuacji bezpieczeństwa
- Należy reagować – nie należy ukrywać błędu!

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

408

Dziękuję za uwagę

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Wykład nr 10
Temat: Podstawy budowy interfejsu użytkownika

Paweł Karczmarek
Katedra Inteligencji Obliczeniowej Politechniki Lubelskiej

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Plan prezentacji

- Wprowadzenie
- Przykłady

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

411

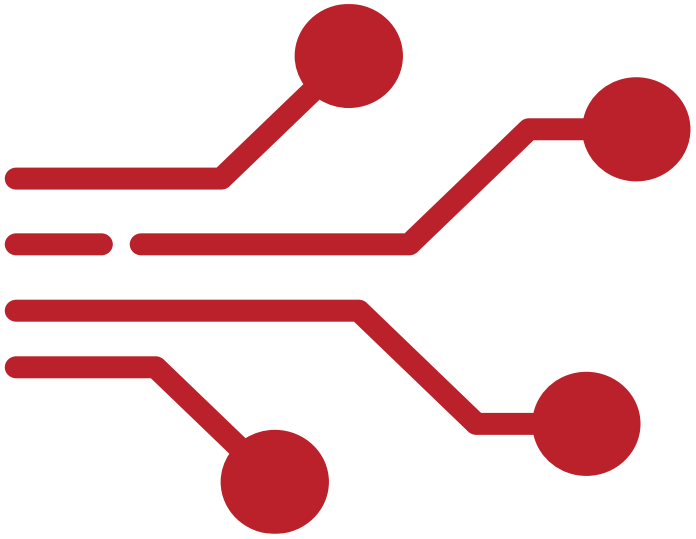
Wprowadzenie

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

412

Wprowadzenie do interfejsów użytkownika (UI)



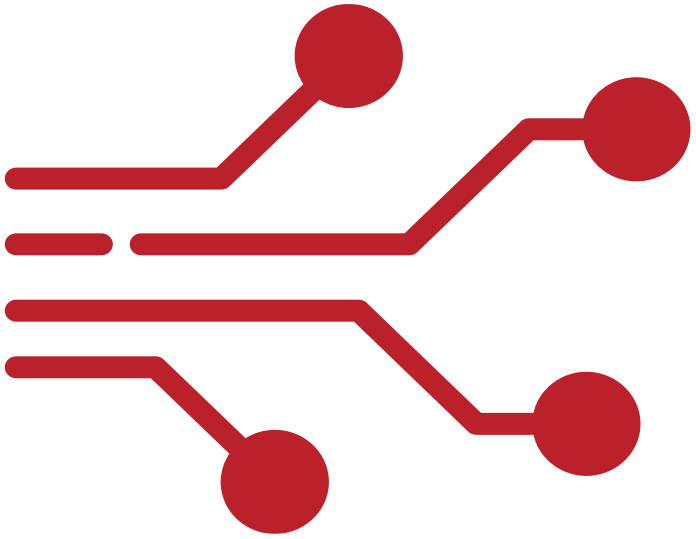
- Interfejs użytkownika (UI) – warstwa, przez którą człowiek komunikuje się z programem
- Dobrze zaprojektowany ma cechy intuicyjności, użyteczności i bezpieczeństwa
- Wyróżniamy interfejsy tekstowe, graficzne, webowe i dotykowe
- W C#: WinForms, WPF, narzędzia firm trzecich (np. Telerik UI)
- Interfejs to nie tylko ładny wygląd, ale również pierwsza linia obrony przed błędami i atakami

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Typy interfejsów: tekstowy, graficzny, webowy

- Interfejs tekstowy (CLI) – komunikacja przez komendy i tekst
- Interfejs graficzny (GUI) – okna, przyciski, pola, kontrolki
- Interfejs webowy – dostępny przez przeglądarkę internetową
- Wszystkie typy interfejsów muszą być użyteczne i bezpieczne
- W .NET: aplikacje konsolowe, WinForms/WPF, ASP.NET/Blazor



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

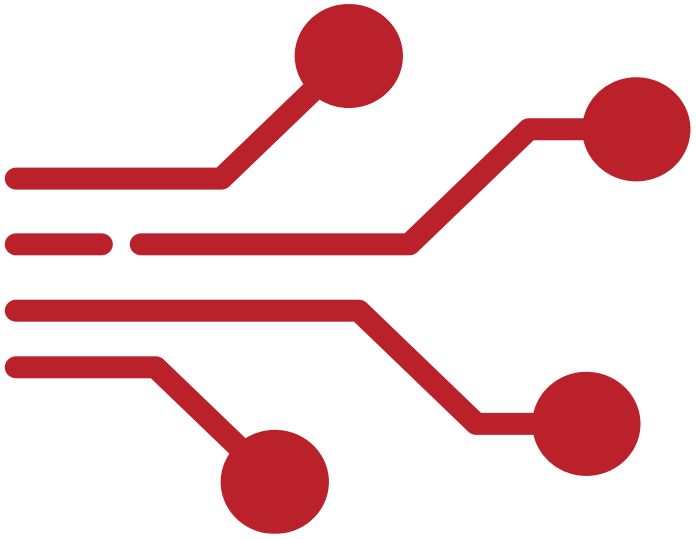
414

Przykłady

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

415



Środowisko Windows Forms (WinForms)

- To klasyczne środowisko tworzenia aplikacji okienkowych w C#
- Formularz (Form, formatka) to główne okno aplikacji
- Interfejs budujemy z gotowych kontrolek: przyciski, pola, etykiety, listy
- Program reaguje na zdarzenia użytkownika (kliknięcie, wpisanie tekstu itp.)
- WinForms daje szybki start, stabilność, prostotę i dobrą bazę dla nauki

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Przykład

```
using System;
using System.Windows.Forms;

class Program
{
    static void Main()
    {
        Application.Run(new Form1());
    }
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

417

Przykład

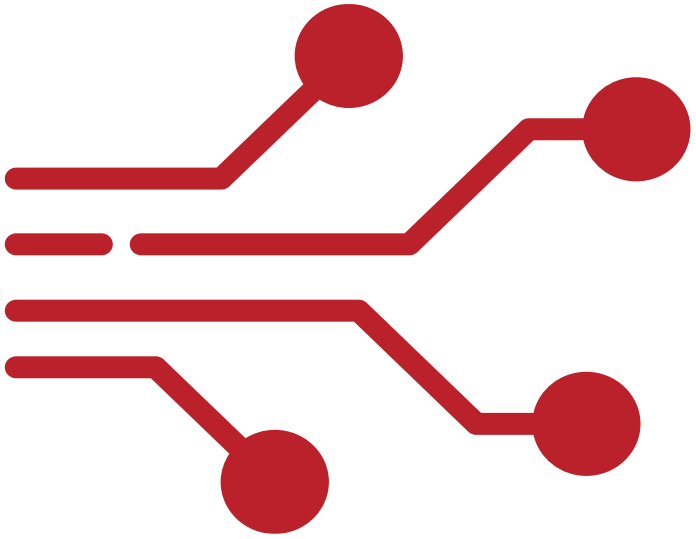
```
public class Form1 : Form
{
    public Form1()
    {
        this.Text = "Pierwsze okno";      // tytuł okna
        this.Width = 400;                  // szerokość
        this.Height = 300;                 // wysokość
        this.BackColor = System.Drawing.Color.LightGray;
            // tło
    }
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

418

Aplikacja okienkowa



- Aplikacja WinForms składa się z formularza i kontrolek (np. Button, Label)
- Każda kontrolka ma właściwości, zdarzenia i metody
- Użytkownik wchodzi w interakcję z GUI (kliknięcia, wpisywanie tekstu)
- Program reaguje na akcje użytkownika za pomocą zdarzeń (Click, TextChanged, …)
- Tworzenie GUI oznacza łączenie logiki z wizualną reprezentacją

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

419

Przykład - prosty formularz z przyciskiem

```
using System;
using System.Windows.Forms;
class Program
{
    [STAThread] // wymóg dla aplikacji GUI w .NET
    static void Main()
    {
        Application.EnableVisualStyles();
        Application.Run(new Form1());
    }
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

420

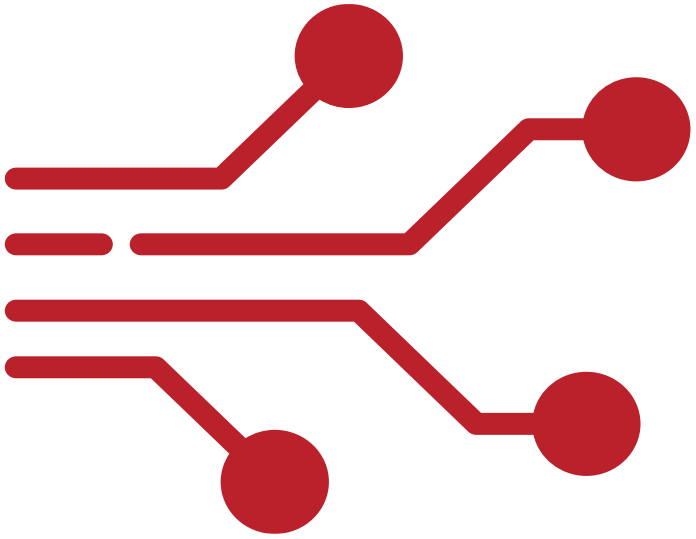
Przykład - prosty formularz z przyciskiem

```
public class Form1 : Form
{
    private Button buttonHello;
    private Label labelMessage;
    public Form1()
    {
        // ustawienia formularza
        this.Text = "Moja pierwsza aplikacja okienkowa";
        this.Width = 400;
        this.Height = 200;
        this.StartPosition = FormStartPosition.CenterScreen;
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Przykład - prosty formularz z przyciskiem

```
// etykieta
labelMessage = new Label();
labelMessage.Text = "Kliknij przycisk!";
labelMessage.Left = 120;
labelMessage.Top = 50;
labelMessage.AutoSize = true;
this.Controls.Add(labelMessage);
```

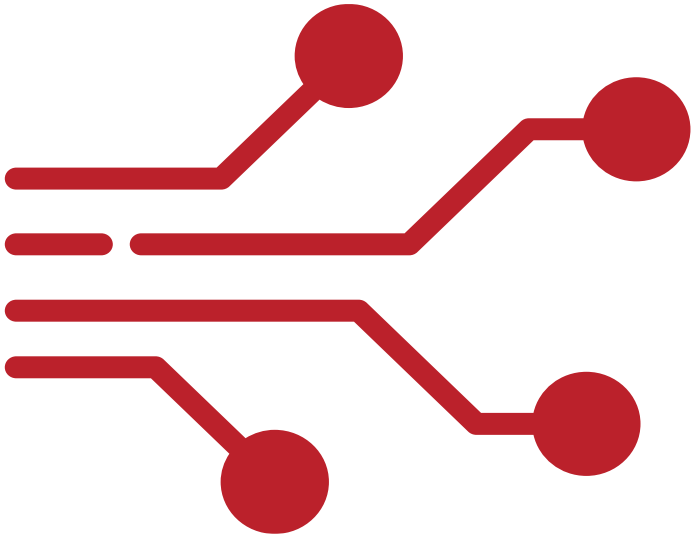


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

422

Przykład - prosty formularz z przyciskiem

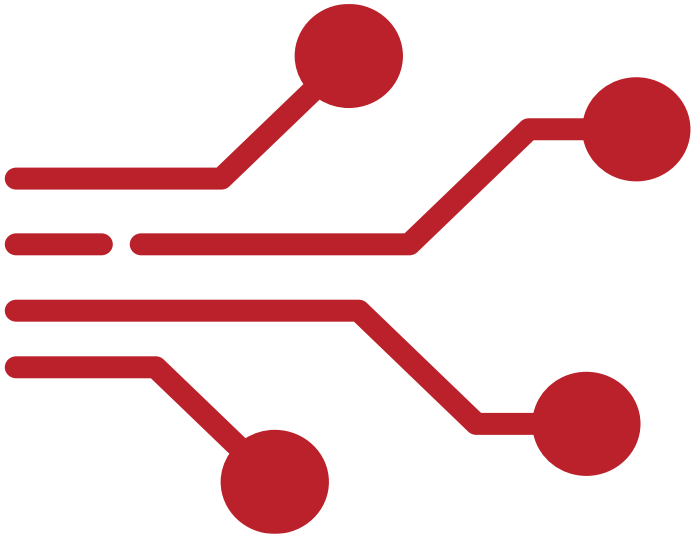
```
            // przycisk
            buttonHello = new Button();
            buttonHello.Text = "Kliknij mnie";
            buttonHello.Left = 130;
            buttonHello.Top = 100;
            buttonHello.Click += ButtonHello_Click;
                // subskrypcja zdarzenia
            this.Controls.Add(buttonHello);
        }
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Przykład - prosty formularz z przyciskiem

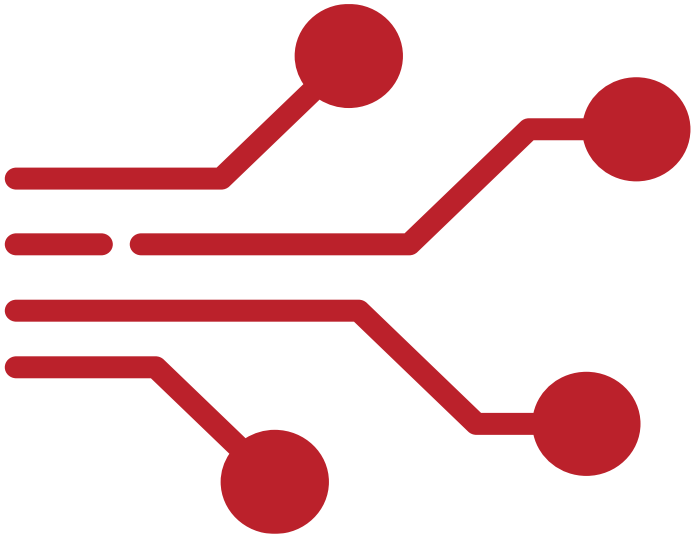
```
private void ButtonHello_Click(object? sender, EventArgs e)
    {
        labelMessage.Text = "Witaj w świecie!";
        labelMessage.ForeColor =
            System.Drawing.Color.DarkBlue;
    }
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

424

Zdarzenia w GUI – interakcja z użytkownikiem



- Zdarzenie (event) to reakcja programu na działanie użytkownika
- Przykłady: kliknięcie przycisku, wpisanie tekstu, zmiana opcji
- Każda kontrolka w WinForms ma własne zdarzenia (Click, TextChanged, Load itd.)
- Program „nasłuchuje” zdarzeń i reaguje metodami obsługi (ang. event handlers)
- W C#: kontrolka.Zdarzenie += MetodaObsługi;

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

425

Przykład – reagowanie na zdarzenia

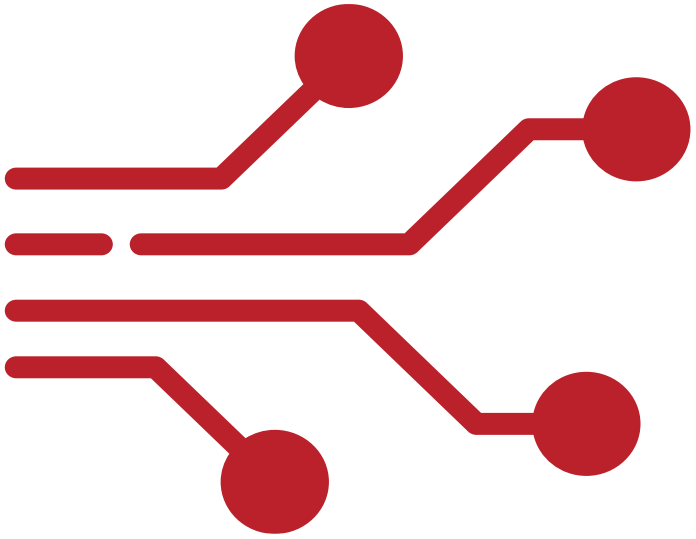
```
using System;
using System.Windows.Forms;
public class Form1 : Form
{
    private TextBox textBoxName;
    private Button buttonShow;
    public Form1()
    {
        this.Text = "Reakcja na zdarzenia";
        this.Width = 350;
        this.Height = 200;
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

426

Przykład – reagowanie na zdarzenia

```
// pole tekstowe
textBoxName = new TextBox();
textBoxName.Left = 50;
textBoxName.Top = 30;
textBoxName.Width = 200;
textBoxName.TextChanged += TextBoxName_TextChanged;
     // zdarzenie wpisania tekstu
this.Controls.Add(textBoxName);
```

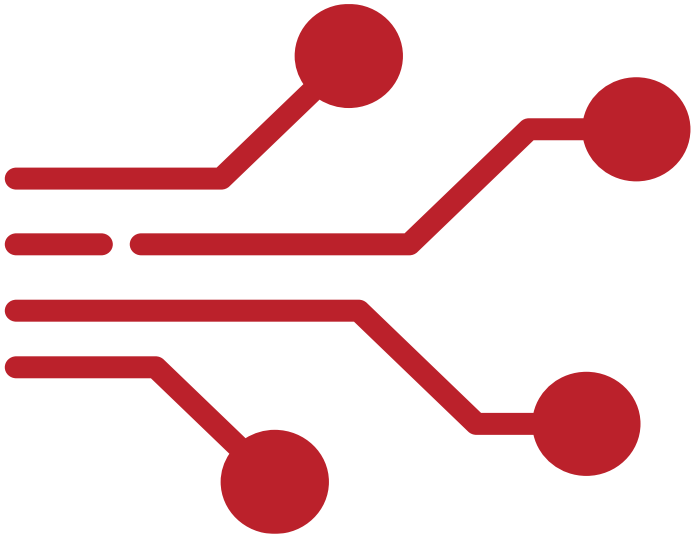


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

427

Przykład – reagowanie na zdarzenia

```
            // przycisk
            buttonShow = new Button();
            buttonShow.Text = "Pokaż imię";
            buttonShow.Left = 100;
            buttonShow.Top = 80;
            buttonShow.Click += ButtonShow_Click;
                // zdarzenie kliknięcia
            this.Controls.Add(buttonShow);
        }
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

428

Przykład – reagowanie na zdarzenia

```
private void TextBoxName_TextChanged(object? sender,
EventArgs e)
{
    this.Text = $"Wpisano: {textBoxName.Text}";
}


private void ButtonShow_Click(object? sender, EventArgs e)
{
    MessageBox.Show($"Cześć, {textBoxName.Text}!",
       "Powitanie");
}
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

429

Przykład prostego formularza logowania

- Formularz logowania = klasyczny przykład interfejsu z walidacją danych
- Składa się z pól: Login, Hasło i przycisku Zaloguj
- Zdarzenia: kliknięcie przycisku, zmiana tekstów
- Maskowanie hasła (PasswordChar = '*') chroni dane na ekranie
- W bezpiecznym UI: brak szczegółowych komunikatów o błędach

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Przykład

```
using System;
using System.Windows.Forms;
public class LoginForm : Form
{
    private Label labelLogin, labelPass;
    private TextBox textLogin, textPass;
    private Button buttonLogin;
    public LoginForm()
    {
        this.Text = "Logowanie do systemu";
        this.Width = 300;
        this.Height = 220;
        this.StartPosition = FormStartPosition.CenterScreen;
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

431

Przykład

```
// etykiety
labelLogin = new Label() { Text = "Login:", Left = 30, Top = 30 };
labelPass  = new Label() { Text = "Hasło:", Left = 30, Top = 70 };
// pola tekstowe
textLogin = new TextBox() { Left = 100, Top = 25, Width = 140 };
textPass  = new TextBox() { Left = 100, Top = 65, Width = 140,
     PasswordChar = '*' };
// przycisk
buttonLogin = new Button() { Text = "Zaloguj", Left = 100, Top =
     110, Width = 100 };
buttonLogin.Click += ButtonLogin_Click;
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

432

Przykład

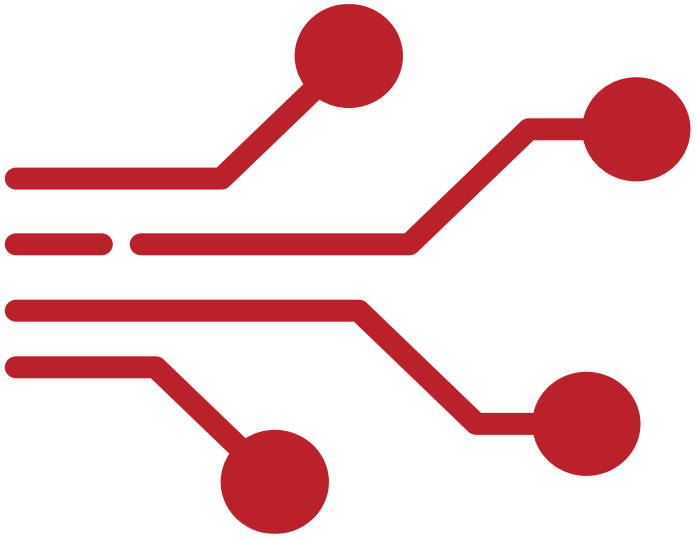
```
    // dodanie elementów do formularza
    Controls.AddRange(new Control[] { labelLogin, textLogin,
         labelPass, textPass, buttonLogin });
}
private void ButtonLogin_Click(object? sender, EventArgs e)
{
    string login = textLogin.Text.Trim();
    string pass  = textPass.Text.Trim();
    if (string.IsNullOrEmpty(login) || string.IsNullOrEmpty(pass))
    {
        MessageBox.Show("Wprowadź login i hasło.", "Błąd logowania",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

433

Przykład

```
        // uproszczona walidacja
        if (login == "admin" && pass == "1234")
            MessageBox.Show("Zalogowano pomyślnie!", "Sukces",
                MessageBoxButtons.OK, MessageBoxIcon.Information);
        else
            MessageBox.Show("Błąd logowania. Sprawdź dane.", "Odmowa
             dostępu",MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

434

Obsługa zdarzeń

- Każda kontrolka (Button, TextBox, CheckBox, ComboBox, itp.) może emitować zdarzenia
- Program może reagować na wiele różnych akcji użytkownika
- Zdarzenia są powiązane z metodami obsługi (event handlers)
- Metody te mają zawsze dwa parametry: object sender, EventArgs e
- Dobre praktyki: krótkie metody obsługi, walidacja danych, logika poza GUI

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

435

Przykład – reagowanie na różne zdarzenia

```
using System;
using System.Windows.Forms;
public class EventDemo : Form
{
    private TextBox inputBox;
    private CheckBox checkConfirm;
    private Button buttonSend;
    public EventDemo()
    {
        this.Text = "Obsługa zdarzeń w praktyce";
        this.Width = 350;
        this.Height = 200;
```

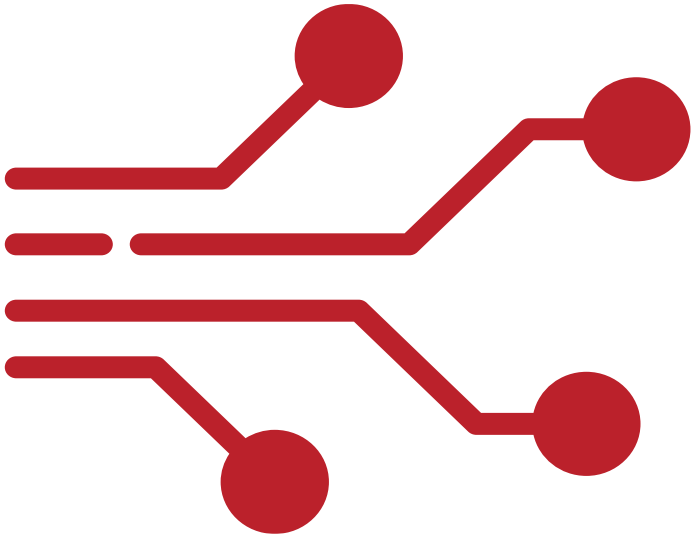

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

436

Przykład – reagowanie na różne zdarzenia

```
// pole tekstowe
inputBox = new TextBox() { Left = 40, Top = 30, Width = 250 };
inputBox.TextChanged += InputBox_TextChanged;


// checkbox
checkConfirm = new CheckBox() { Text = "Potwierdzam
      wysłanie", Left = 40, Top = 70 };
checkConfirm.CheckedChanged += CheckConfirm_CheckedChanged;
```



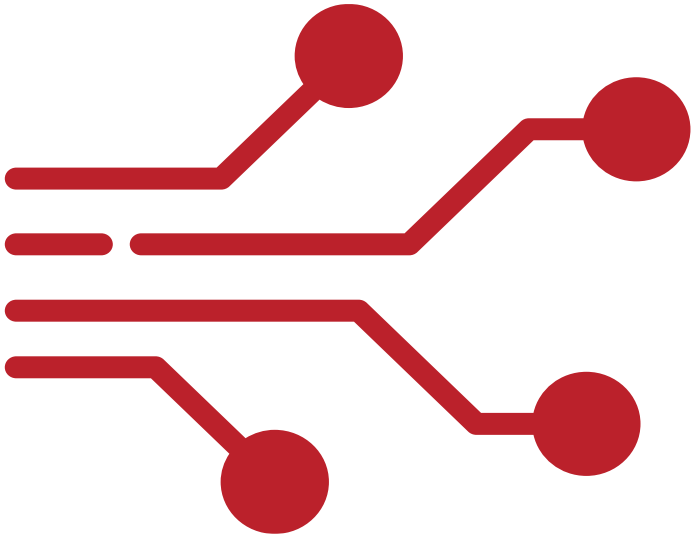
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

437

Przykład – reagowanie na różne zdarzenia

```
            // przycisk
            buttonSend = new Button() { Text = "Wyślij", Left = 120, Top
                  = 110, Enabled = false };
            buttonSend.Click += ButtonSend_Click;


            Controls.AddRange(new Control[] { inputBox, checkConfirm,
                  buttonSend });
      }
```



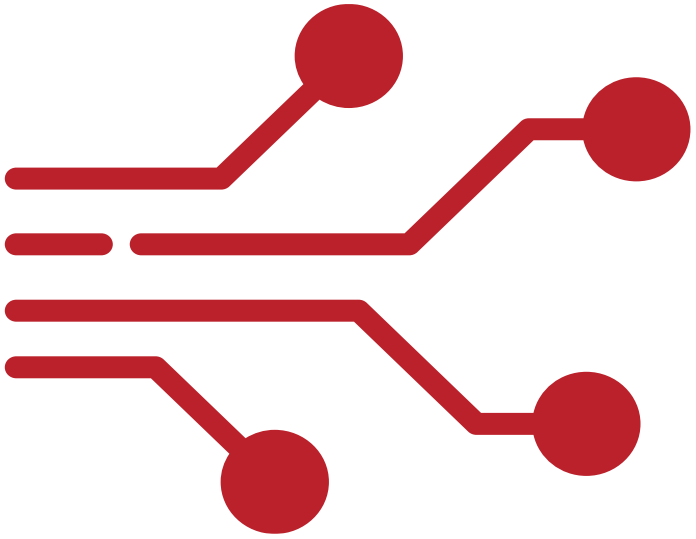
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

438

Przykład – reagowanie na różne zdarzenia

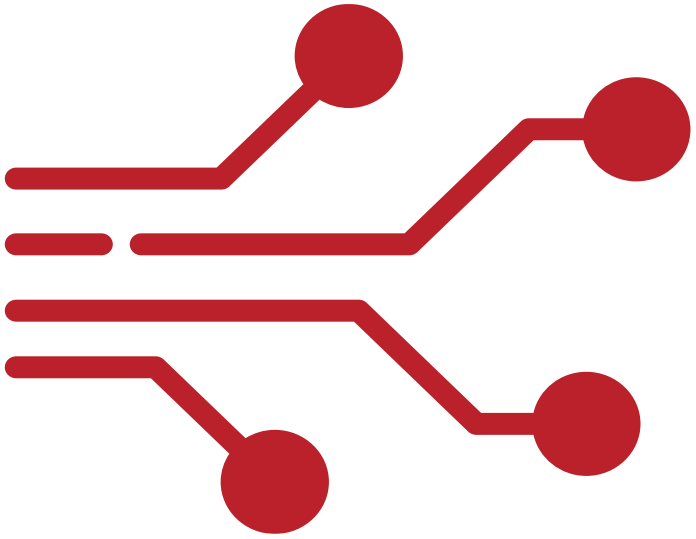
```
private void InputBox_TextChanged(object? sender, EventArgs e)
{
    Console.WriteLine($"📥 Wpisano: {inputBox.Text}");
}


private void CheckConfirm_CheckedChanged(object? sender,
    EventArgs e)
{
    buttonSend.Enabled = checkConfirm.Checked;
          // aktywuj przycisk tylko po potwierdzeniu
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Przykład – reagowanie na różne zdarzenia

```
    private void ButtonSend_Click(object? sender, EventArgs e)
    {
        MessageBox.Show($"Wiadomość '{inputBox.Text}' została
              wysłana.", "Sukces");
    }
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

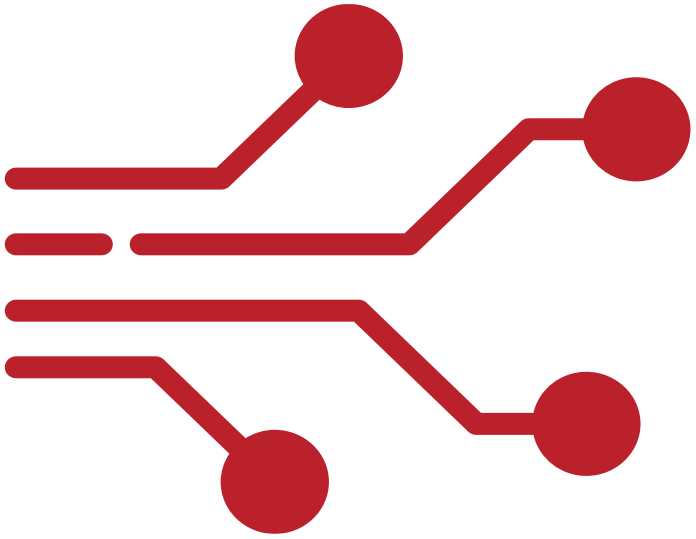
440

Najczęstsze błędy w interfejsach użytkownika

- Brak walidacji danych wprowadzanych przez użytkownika
- Zbyt szczegółowe komunikaty o błędach (ujawnianie informacji)
- Niejasny lub mylący interfejs – użytkownik wykonuje błędne akcje
- Elementy aktywne mimo braku uprawnień
- Niechronione pola – dane jawne, hasła widoczne na ekranie

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Dobre praktyki bezpieczeństwa w interfejsach użytkownika (UI)



- Walidacja wszystkich danych wejściowych – nigdy nie ufamy użytkownikowi
- Maskowanie danych wrażliwych (haseł, numerów, kluczy)
- Unikanie ujawniania szczegółowych błędów
- Oddzielenie logiki biznesowej od warstwy interfejsu
- Ukrywanie elementów nieprzeznaczone dla danego użytkownika
- Zapewnienie jednoznacznego, przewidywalnego przepływu interakcji

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

442

Przykład

```
private void ButtonSend_Click(object? sender, EventArgs e)
{
    string email = textEmail.Text.Trim();
    if (string.IsNullOrEmpty(email))
    {
        MessageBox.Show("Pole e-mail nie może być puste.", "Błąd",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    int atPos = email.IndexOf('@');
    int dotPos = email.LastIndexOf('.');
```

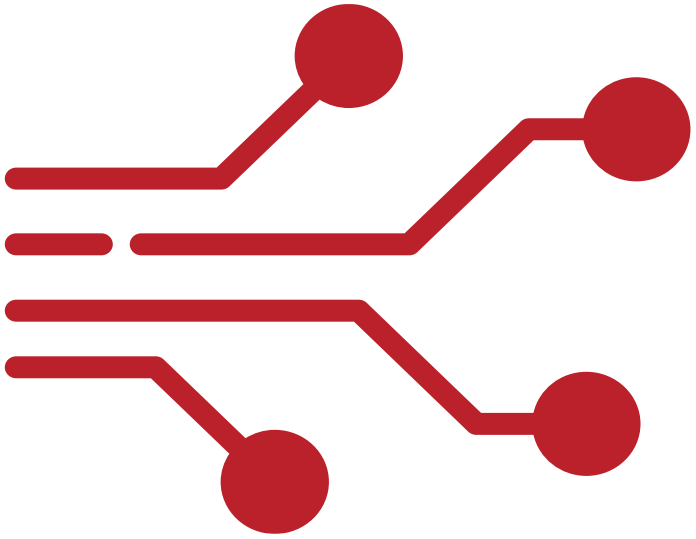

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

443

Przykład

```
    if (atPos < 1 || dotPos < atPos + 2 || dotPos == email.Length - 1)
        {
            MessageBox.Show("Podaj poprawny adres e-mail.", "Błąd",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
            return;
        }

        MessageBox.Show("Dane wysłane pomyślnie!", "Sukces",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
}
```



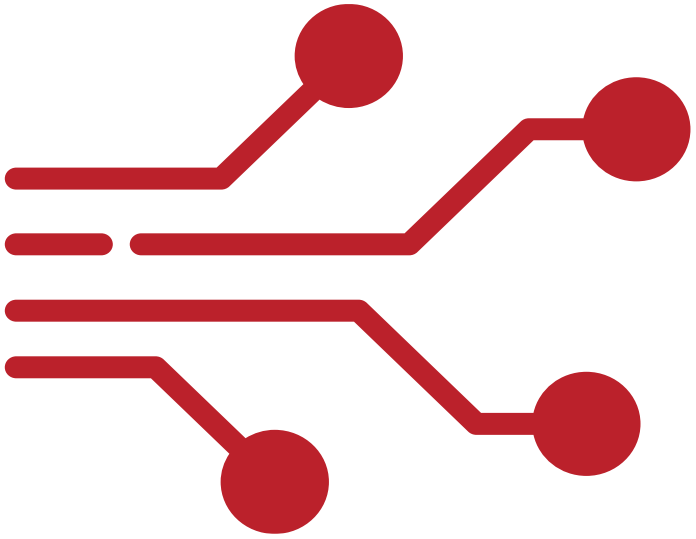
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

444

Wariant

```
var parts = email.Split('@');
if (parts.Length != 2 || !parts[1].Contains('.'))
{
    MessageBox.Show("Niepoprawny format adresu e-mail.", "Błąd",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    return;
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

445

Uwaga

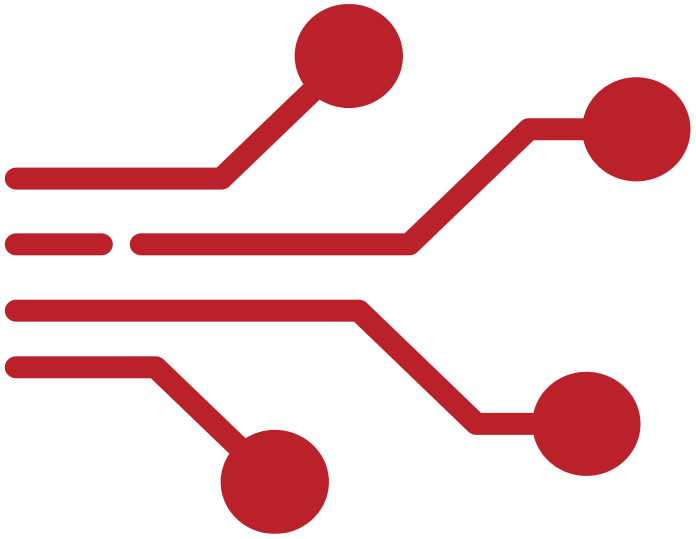
Właściwości kontrolek możemy ustawiać w oknie Properties (podobnie zdarzenia – Events), natomiast same kontrolki możemy przeciągać z przybornika (Toolbox). Klikając w kontrolki na formatce możemy tworzyć lub przechodzić do odpowiednich fragmentów kodu.

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

446

WPF (Windows Presentation Foundation)



- WPF (Windows Presentation Foundation) – nowoczesny framework do budowy interfejsów graficznych w C#
- Oddziela warstwę wizualną (XAML) od logiki programu (C#)
- Umożliwia tworzenie złożonych, dynamicznych i bezpiecznych interfejsów
- Obsługuje stylizację, bindowanie danych, animacje, szablony, MVVM
- Jest częścią platformy .NET

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

447

Przykład: MainWindow.xaml

```
<Window x:Class="HelloWPF.MainWindow"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="WPF Demo" Height="200" Width="350">
    <StackPanel VerticalAlignment="Center">
        <TextBlock Text="Podaj imię:" Margin="10" FontSize="14"/>
        <TextBox Name="txtName" Width="200" Margin="10"/>
        <Button Content="Powitaj" Width="100" Margin="10"
Click="Button_Click"/>
    </StackPanel>
</Window>
```

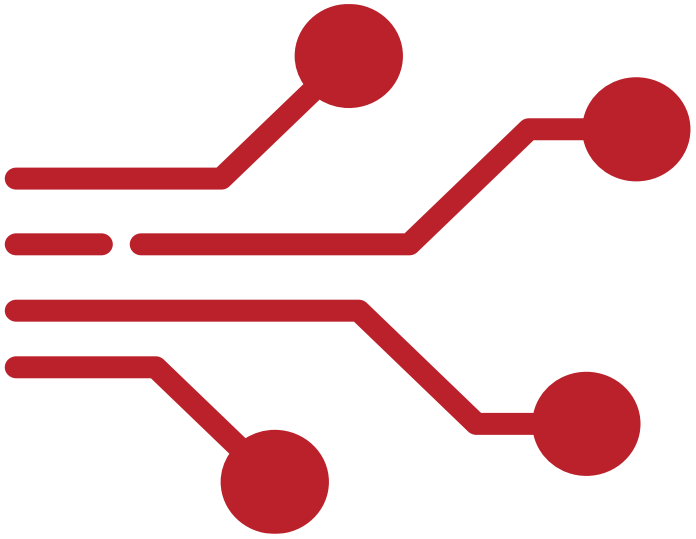

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

448

Przykład: MainWindow.xaml.cs

```
using System.Windows;
namespace HelloWPF
{
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

449

Przykład: MainWindow.xaml.cs

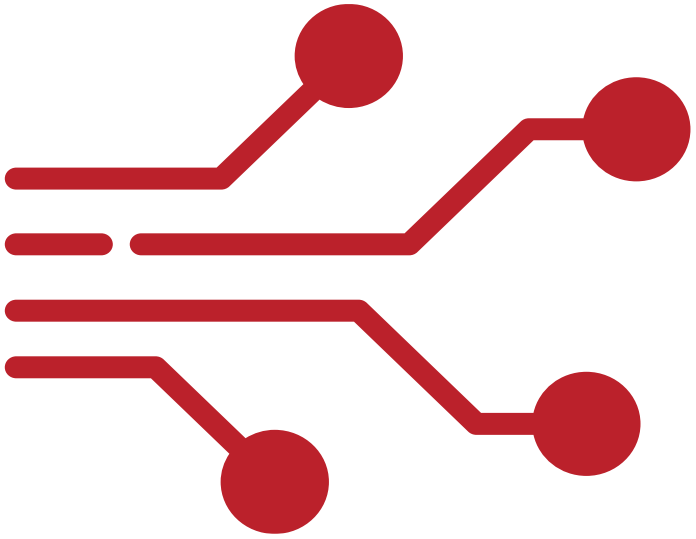
```
        private void Button_Click(object sender,
            RoutedEventArgs e)
        {
            MessageBox.Show($"Witaj, {txtName.Text}!",
             "Powitanie");
        }
    }
}
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

450

Telerik UI i komponenty zewnętrzne

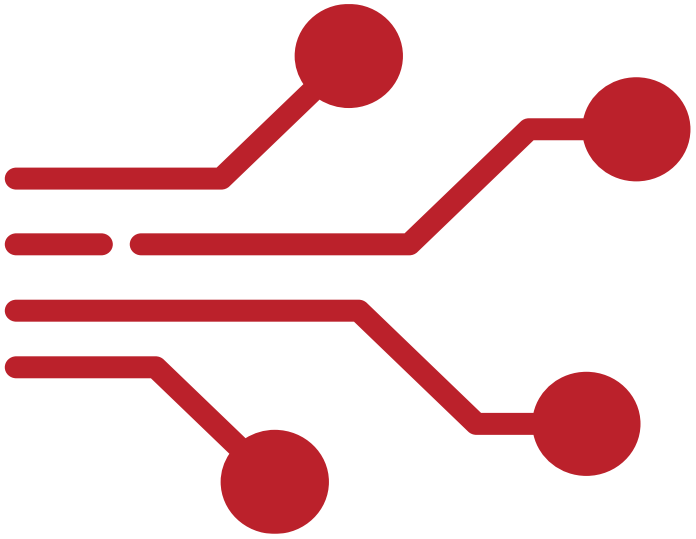


- W nowoczesnych projektach rzadko buduje się GUI od zera
- Wykorzystuje się gotowe biblioteki: Telerik UI, DevExpress, Syncfusion
- Oferują profesjonalne kontrolki: siatki danych, wykresy, kalendarze, panele, dashboardy
- Ułatwiają tworzenie interfejsu spójnego, wydajnego i nowoczesnego
- Wymagają jednak ostrożności: licencje, aktualizacje, zaufane źródła

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

451



Integracja interfejsu z logiką aplikacji

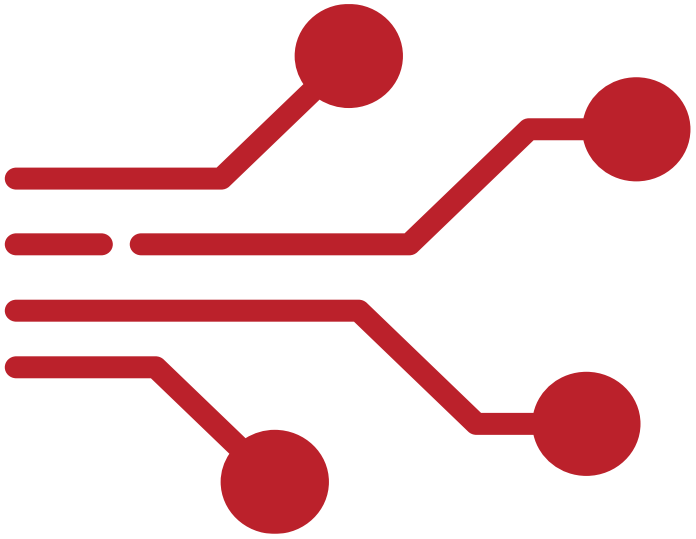
- Interfejs użytkownika (UI) powinien komunikować się z logiką programu, ale jej nie zawierać
- Logika programu powinna być w osobnych klasach/warstwach (np. Service, Model)
- UI wywołuje logikę przez metody, zdarzenia lub wzorce projektowe (MVC – Model View Controller /MVVM – Model – View - ViewModel)
- Celem jest: separacja, czytelność, testowalność i bezpieczeństwo
- Przykład: formularz logowania wywołuje klasę AuthService, a nie zawiera logiki w sobie

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

452

Przykład – separacja interfejsu i logiki

```
// Logika programu (oddzielna klasa)
public class AuthService
{
    public bool Login(string username, string password)
    {
        // Tu tylko logika, bez GUI!
        return username == "admin" && password == "1234";
    }
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

453

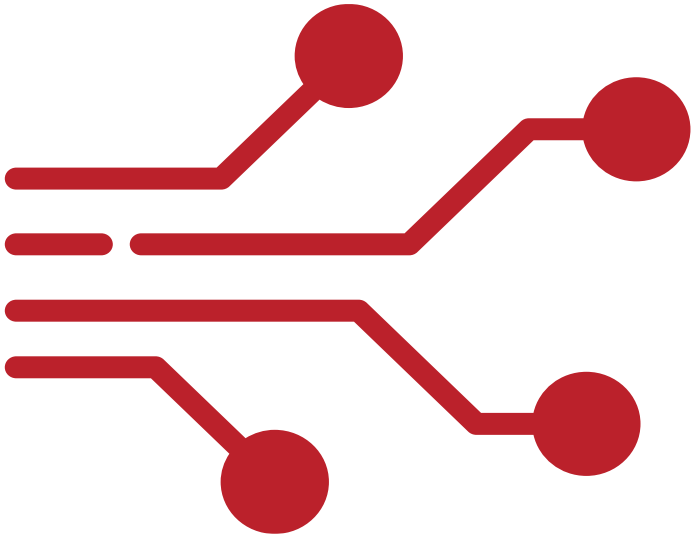
Przykład – separacja interfejsu i logiki

```
// Interfejs użytkownika (Formularz)
public partial class LoginForm : Form
{
    private AuthService authService = new AuthService();
    public LoginForm()
    {
        InitializeComponent();
    }
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Przykład – separacja interfejsu i logiki

```
    private void buttonLogin_Click(object sender, EventArgs e)
    {
        string user = textBoxUser.Text.Trim();
        string pass = textBoxPass.Text.Trim();

        if (authService.Login(user, pass))
            MessageBox.Show("Zalogowano pomyślnie!");
        else
            MessageBox.Show("Błąd logowania.");
    }
}
```



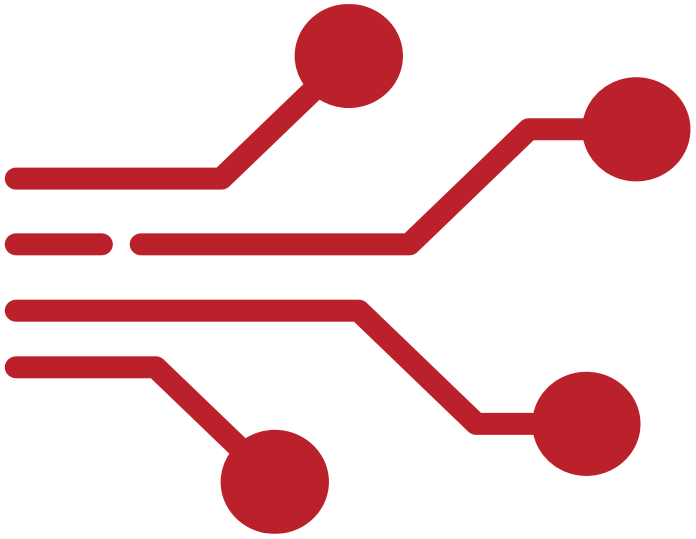
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

455

Zatrzymywanie akcji

- Niektóre zdarzenia w WinForms kończą się na -ing (np. FormClosing, Validating, KeyPress)
- Oznacza to, że akcja jeszcze się nie wykonała i można ją zatrzymać
- Zdarzenia te służą do walidacji i zapobiegania błędom użytkownika
- Można ustawić e.Cancel = true, by przerwać operację
- Idealne narzędzie do potwierdzeń i zabezpieczeń w GUI

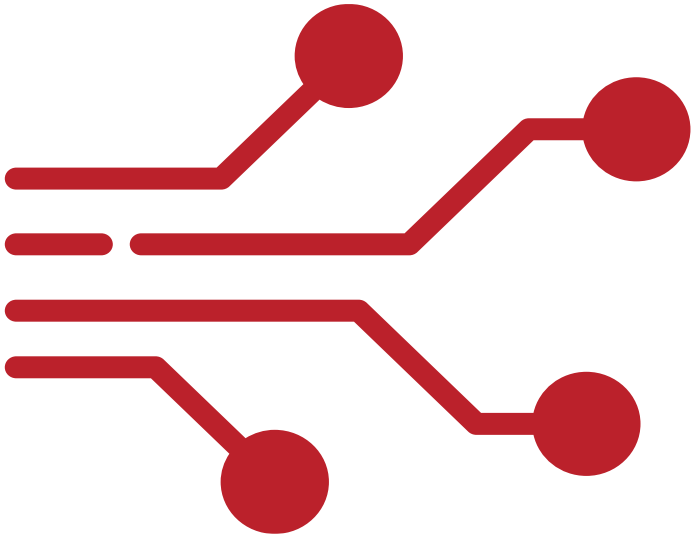


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

456

Przykład

```
public partial class SecureForm : Form
{
    private TextBox textBoxEmail = new TextBox();
    public SecureForm()
    {
        this.Text = "Formularz z walidacją";
        this.Controls.Add(textBoxEmail);
        textBoxEmail.Top = 30;
        textBoxEmail.Width = 200;
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

457

Przykład

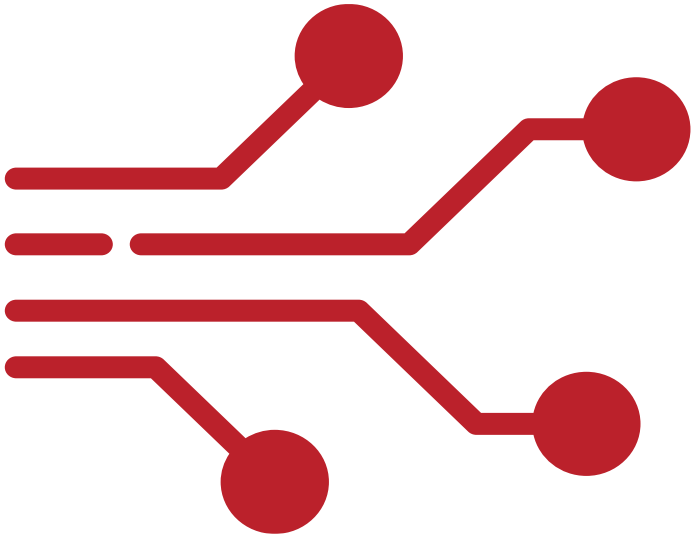
```
// potwierdzenie zamknięcia okna
this.FormClosing += (s, e) =>
{
    var result = MessageBox.Show("Czy na pewno chcesz
       zamknąć okno?",
                                 "Potwierdzenie",
                                 MessageBoxButtons.YesNo,
                                 MessageBoxIcon.Question);
    if (result == DialogResult.No)
        e.Cancel = true; // zatrzymaj zamykanie
};
```


Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Przykład

```
// walidacja pola
textBoxEmail.Validating += (s, e) =>
{
    if (!textBoxEmail.Text.Contains("@"))
    {
        MessageBox.Show("Niepoprawny e-mail!");
        e.Cancel = true; // zatrzymaj przejście dalej
    }
};
```



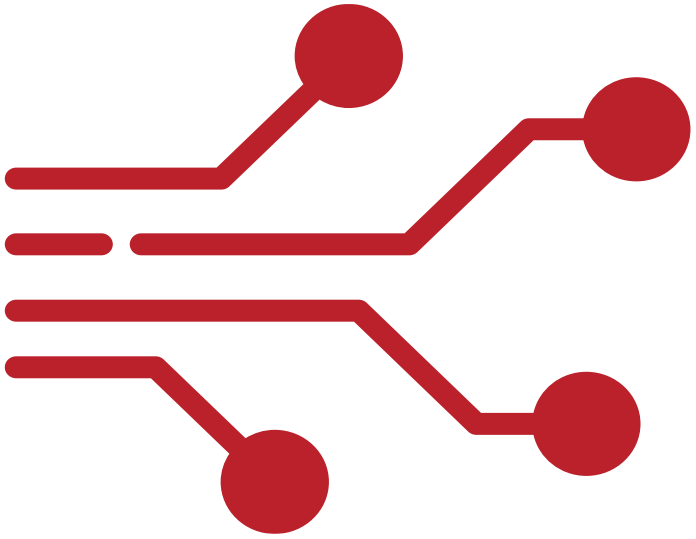
Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

459

Przykład

```
            // filtr klawiatury
            textBoxEmail.KeyPress += (s, e) =>
            {
                if (char.IsWhiteSpace(e.KeyChar))
                    e.Handled = true; // zablokuj spację
            };
        }
}
```



Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

460

Dziękuję za uwagę

Projekt współfinansowany ze środków Unii Europejskiej w ramach:
FUNDUSZE EUROPEJSKIE DLA ROZWOJU SPOŁECZNEGO 2021-2027 (FERS)
"POLLUB z nami nowoczesne technologie" FERS.01.05-IP.08-0319/23-00