



## **Programowanie obiektowe 2023/2024**

### **Laboratorium 1**

**Praca w środowisku Visual Studio, tworzenie prostych  
programów w języku C#**



## Wstęp

<https://learn.microsoft.com/pl-pl/visualstudio/ide/?view=vs-2022>

## Podstawowe elementy składni języka C#

**Komentarze** - Możesz dodawać komentarze do kodu, które nie będą wykonywane, ale pomogą w jego zrozumieniu.

```
// To jest komentarz jednoliniowy
```

```
/* To jest komentarz
```

```
wieloliniowy */
```

**Zmienne** - W C# deklarujesz zmienne z określonym typem danych.

```
int liczbaCalkowita = 42;  
double liczbaZmiennoprzecinkowa = 3.14;  
string napis = "Hello, World!";  
var liczba = 43; //int
```

**Instrukcje warunkowe** - Możesz używać instrukcji warunkowych do sterowania przepływem programu.

```
if (warunek)  
{  
    // Wykonaj kod, jeśli warunek jest spełniony  
}  
else if (innyWarunek)  
{  
    // Wykonaj kod, jeśli inny warunek jest spełniony  
}  
else  
{  
    // Wykonaj kod, jeśli żaden z powyższych warunków nie jest spełniony  
}
```

**Pętle** - Pętle pozwalają na wielokrotne wykonanie kodu.

```
for (int i = 0; i < 5; i++)  
{  
    // Wykonaj kod 5 razy  
}
```

```
while (warunek)  
{  
    // Wykonuj kod, dopóki warunek jest spełniony  
}
```

```
do  
{  
    // Wykonuj kod przynajmniej raz, a potem sprawdzaj warunek  
} while (warunek);
```



**Funkcje** - Funkcje to bloki kodu, które można wywoływać w różnych miejscach programu.

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

```
// Wywołanie funkcji
int wynik = Dodaj(2, 3);
```

**Tablice** - Tablice pozwalają na przechowywanie wielu elementów tego samego typu.

```
int[] liczby = { 1, 2, 3, 4, 5 };
string[] dniTygodnia = new string[7];
dniTygodnia[0] = "Poniedziałek";
```

**Klasy i obiekty** - Klasy są szablonami do tworzenia obiektów. Obiekt to instancja klasy.

```
class Osoba
{
    public string Imie;
    public string Nazwisko;
}
```

```
Osoba osoba1 = new Osoba();
osoba1.Imie = "Jan";
osoba1.Nazwisko = "Kowalski";
```

**Operatory** - C# posiada wiele operatorów, takich jak +, -, \*, /, ==, !=, itp., które służą do wykonywania operacji na zmiennych.

```
int wynikDodawania = 2 + 3;
bool czyRowne = wynikDodawania == 5;
```

**Wyjątki** - W C# można obsługiwać wyjątki za pomocą bloków try-catch.

```
try
{
    // Kod, który może spowodować wyjątek
}
catch (Exception ex)
{
    // Obsługa wyjątku
    Console.WriteLine("Wystąpił wyjątek: " + ex.Message);
}
```



## **Struktura aplikacji konsolowej**

Aplikacja konsolowa w C# składa się z klasy z metodą Main, która jest punktem wejścia do programu.

```
using System;
```

```
class Program  
{  
    static void Main()  
    {  
        // Kod programu  
    }  
}
```

Do wyświetlania tekstu można używać metody **Console.WriteLine**.

```
Console.WriteLine("Witaj, to jest aplikacja konsolowa!");
```

W celu wczytania danych od użytkownika można zastosować **Console.ReadLine**.

```
string imie = Console.ReadLine();
```



## **Zadania**

**Zadanie 1.** Napisz program, który sprawdzi, czy podana przez użytkownika liczba jest parzysta czy nieparzysta.

**Zadanie 2.** Napisz program, który wypisze na konsoli wszystkie parzyste liczby od 1 do N, gdzie N jest liczbą wprowadzoną przez użytkownika.

**Zadanie 3.** Napisz program, który wyświetli „menu” użytkownika. Po wybraniu pozycji z menu (poprzez podanie odpowiedniej liczby) program wykona odpowiednio przypisaną funkcję z pozostałych zadań.

**Zadanie 4.** Napisz program, który obliczy silnie ze wskazanej przez użytkownika liczby.

**Zadanie 5.** Napisz grę, w której komputer losuje liczbę, a użytkownik próbuje odgadnąć ją w jak najmniejszej liczbie prób.

**Zadanie 6.** Napisz program, które umożliwia przeliczanie jednostek miar, takich jak przeliczanie temperatury między stopniami Celsjusza a stopniami Fahrenheita lub między jednostkami długości, np. metrami i centymetrami.