



Politechnika
Wroclawska

Platformy programistyczne

Typy danych i elementy programowania obiektowego

Dr inż. Radosław Idzikowski

Katedra Automatyki, Mechatroniki i Systemów Sterowania
Wydział Informatyki i Teleinformatyki

12 marca 2024



Rodzaje typów

- ▶ wartościowe:
 - ▶ większość typów wbudowanych (liczbowe, char i bool),
 - ▶ zdefiniowane struktury (struct),
 - ▶ wyliczenia (enum).
- ▶ referencyjne
 - ▶ klasy,
 - ▶ tablice,
 - ▶ interfejsy,
 - ▶ predefiniowany typ string oraz object.

Podstawowa różnica w przechowywaniu ich wartości w pamięci!



Typy wartościowe

- ▶ przechowują wprost swoją wartość.
- ▶ przypisanie instancji typu wartościowego zawsze powoduje jego skopiowanie.

```
class Program
{
    public struct Point { public int X, Y; }
    static void Main()
    {
        Point A = new Point();
        A.X = 1;
        Point B = A;
        B.X = 2;
        Console.WriteLine(A.X);
        Console.WriteLine(B.X);
    }
}
```



Typy referencyjne

- ▶ zawartość jego zmiennej lub stałej stanowi referencja do obiektu zawierającego wartość,
- ▶ przypisanie zmiennej typu ref. spowoduje skopiowanie jego referencji.

```
class Program
{
    public class Point { public int X, Y; }
    static void Main()
    {
        Point A = new Point();
        A.X = 1;
        Point B = A;
        B.X = 2;
        Console.WriteLine(A.X);
        Console.WriteLine(B.X);
    }
}
```



Wartość null

- ▶ zmiennej typu referencyjnego można przypisać wartość null oznaczający, że dana referencja nie wskazuje żadnego obiektu.

```
class Program
{
    public class Point { public int X, Y; }
    static void Main()
    {
        Point A = null;
    }
}
```



Typ string

- ▶ dosłowne literały łańcuchowe.

```
class Program
{
    static void Main()
    {
        string str1 = "Hello\nWorld!";
        string str2 = @"Hello
World!";

        Console.WriteLine(str1);
        Console.WriteLine(str2);
    }
}
```



Typ string

- ▶ konkatencja,
- ▶ interpolacja.

```
class Program
{
    static void Main()
    {
        string str1 = "Hello" + " " + "World!";
        int number = 11;
        string str2 = "Your number: " + number;
        string str3 = $"Your number: {number}!";
        Console.WriteLine(str3);
    }
}
```



Tablice

- ▶ dziedziczenie po klasie `System.Array`,
- ▶ wyrażenie inicjalizacyjne,
- ▶ możliwość przechowywania typów wartościowych oraz referencyjnych,
- ▶ tablica zawsze jest obiektem typu referencyjnego.

```
class Program
{
    struct PointS { public int X, Y; }
    class PointC { public int X, Y; }
    static void Main()
    {
        int [] arr1 = new int []{ 1, 2, 3 };
        int [] arr2 = { 1, 2, 3 };
        PointS [] pointsS = new PointS [100];
        PointC [] pointsC = new PointC [100];
        int [] arr3 = null;
    }
}
```




Tablice dwu- i wielo- wymiarowe

```
class Program
{
    static void Main()
    {
        int[,] A = {
            {1,2,3,4 },
            {5,6,7,8}
        };
        int [][] B = {
            new int []{1,2,3,4 },
            new int []{5,6,7,8}
        };
    }
}
```



Typ object

```
public class Stack
{
    int x;
    object[] objects;
    public Stack(int size = 10)
    { x = 0; objects = new object[size]; }
    public void Push (object obj) { objects[x++] = obj;}
    public object Pop () { return objects[--x]; }
}
static void Main()
{
    Stack stack = new Stack();
    stack.Push(1.5f);
    float number = (float)stack.Pop();
    Console.WriteLine(number);
}
```

Własności

- ▶ poza klasą wyglądają jak pola, ale wewnątrz zawierają logikę, tak jak metody.
- ▶ get i set,
- ▶ sposób używania:
 - ▶ własna implementacja,
 - ▶ wyrażeniowe,
 - ▶ automatyczne.





Windows Forms

- ▶ Form,
- ▶ TextBox,
- ▶ ListBox,
- ▶ Button,
- ▶ oraz wiele innych.

