

## Laboratorium 3

### Programowanie w C# (semestr zimowy 2019/2020)

Temat: Wprowadzenie do programowania w języku C#.

#### Tablice

Prowadzący: Piotr Pięta

## 1. Wprowadzenie

- 1.1. Operator arytmetyczny **inkrementacji (dekrementacji)** występuje w dwóch popularnych odmianach: przedrostkowej (++zmienna) i przyrostkowej (zmienna--). Oba te wyrażenia nie są sobie równoważne.

Operator x++ zwiększa wartość zmiennej (zmiennej x) o jeden, lecz po wykonaniu operacji (wykorzystaniu zmiennej). Przeciwny przypadek – operator ++ przed nazwą zmiennej – zmiana wartości zmiennej następuje przed wykorzystaniem zmiennej.

Odniesienie do dokumentacji, uruchom przykłady dot. operatorów ++/--

<https://docs.microsoft.com/pl-pl/dotnet/csharp/language-reference/operators/arithmetic-operators>

```
static void Main(string[] args)
{
    int x = 0, y;
    y = x++ * 2;
    Console.WriteLine(x); //x <- 1
    Console.WriteLine(y); //y <- 0
    Console.ReadKey();
}

static void Main(string[] args)
{
    int x = 0, y;
    y = ++x * 2;
    Console.WriteLine(x); //x <- 1
    Console.WriteLine(y); //y <- 2
    Console.ReadKey();
}
```

```

static void Main(string[] args)
{
    int x, y = 5; // x <- ? y <-?
    x = ++y; // x <- ? y <-?
    x = y++; // x <- ? y <-?
    x = --y; // x <- ? y <-?
    x = y--; // x <- ? y <-?
    Console.WriteLine(y++); // x <- ? y <-?
    Console.ReadKey();
}

```

//przykład z alternatywą warunkową –  
SKOMENTOWAĆ NA ZAJĘCIACH

```

static void Main(string[] args)
{
    int x = 3, y = 10;
    bool wynik;
    wynik = (x >= 2 || y++ >= 2);
    Console.WriteLine(wynik);
    Console.WriteLine(x);
    Console.WriteLine(y);
    Console.ReadKey();
}

static void Main(string[] args)
{
    int x = 1, y = 2, z = 3;
    bool wynik;
    wynik = (x == 1 || y != 5 && z < 1);
    //wynik = ((x == 1 || y != 5) && z < 1);
    //wynik = !(x == 1 || y != 5 && z < 1);
    Console.WriteLine(wynik);
    Console.ReadKey();
}

```

- 1.2 Wszystkie dane typu string (np. literały znakowe) są **niezmienne**, czyli niemodyfikowalne. Nie można zmienić łańcucha znaków z *Jan Kowalski pilnie się uczy* na *Jan Kowalski uczy się poniżej przeciętnej* bez zmiany polegającej na tym, że zmienna zacznie wskazywać nową lokalizację w pamięci. Nie można zmodyfikować danych pierwotnie wskazywanych przez zmienną
- 1.3 **Interpolacja łańcuchów znaków** (wprowadzona od wersji C# 6.0) – alternatywny sposób wyświetlania wartości. Kompilator za pomocą tego mechanizmu interpretuje zawartość nawiasów klamrowych w łańcuchach znaków jako obszar, w którym można osadzać kod (wyrażenia) przetwarzany przez kompilator i przekształcany na tekst. Zamiast wykonywać osobno wiele fragmentów kodu i na końcu łączyć

wyniki w łańcuchach znaków, za pomocą omawianego mechanizmu można osiągnąć ten sam efekt w jednym kroku. Dzięki temu kod jest bardziej zrozumiały.

Dla zainteresowanych: <https://docs.microsoft.com/pl-pl/dotnet/csharp/tutorials/string-interpolation>

**Prześledź działanie gotowych kodów** (linki odnoszące się do Dokumentacji i materiałów pomocniczych do nauki języka C#).

```
C# Kopiuuj Uruchom
double a = 3;
double b = 4;
Console.WriteLine($"Area of the right triangle with legs of {a} and {b} is {0.5 * a * b}");
Console.WriteLine($"Length of the hypotenuse of the right triangle with legs of {a} and {b} is {CalculateHypotenuse(a, b)}");

double CalculateHypotenuse(double leg1, double leg2) => Math.Sqrt(leg1 * leg1 + leg2 * leg2);

// Expected output:
// Area of the right triangle with legs of 3 and 4 is 6
// Length of the hypotenuse of the right triangle with legs of 3 and 4 is 5
```

Wymagany jest dostęp do www 😊 Aby prześledzić konkretne przypadki (kody) i uzyskać w łatwy i szybki sposób wynik, konieczne kliknij w Uruchom. Po chwili powinien wygenerować się wynik.

```
Edytor technologii .NET Uruchom
1 double a = 3;
2 double b = 4;
3 Console.WriteLine($"Area of the right triangle with legs of {a} and {b} is {0.5 * a * b}");
4 Console.WriteLine($"Jan Kowlski ? {a} and {b} is {CalculateHypotenuse(a, b)}");
5
6 double CalculateHypotenuse(double leg1, double leg2) => Math.Sqrt(leg1 * leg1 + leg2 * leg2);
7
8 // Expected output:
9 // Area of the right triangle with legs of 3 and 4 is 6
10 // Length of the hypotenuse of the right triangle with legs of 3 and 4 is 5
```

Dane wyjściowe

```
Area of the right triangle with legs of 3 and 4 is 6
Jan Kowlski ? 3 and 4 is 5
```

W edytorze możemy (co mam nadzieję, oczywiste dla Państwa) dokonywać modyfikacji i zmian w liniach kodu.

Zachęcam do uruchamiania gotowych przykładów zawartych w Dokumentacji (nie tylko wskazanych w laborce).

## 1.5 Tablice w C#

Struktury w języku C# mogące przechowywać wiele elementów tego samego typu. Dwa ważne pojęcia związane z tablicami to ich **rozmiar** oraz **wymiar**. Wyróżniamy tablice jedno-, dwu-, wielo- wymiarowe mogące pomieścić dane. Szczególnym przypadkiem jest tablica nieregularna. Indeksowanie tablicy rozpoczyna się od 0, a kończy na (n-1), gdzie n-rozmiar (danej) tablicy. Elementy tablicy przyjmują wartości nie tylko podstawowe (int, char etc.), lecz również typy tablicowe.

Ogólny schemat deklaracji tablicy:

```
typ_danych[] nazwa_tablicy;  
//[ ] – definiuje rozmiar
```

Podstawy, intuicyjne przykłady:

[https://www.plukasiewicz.net/CSharp\\_dla\\_poczatkujacych/Tablice](https://www.plukasiewicz.net/CSharp_dla_poczatkujacych/Tablice)

*Przykład:*

// Deklaracja tablicy jednowymiarowej wraz z przypisaniem referencji

```
int[] tablica1 = new int[5];
```

// Deklaracja tablicy wraz z dokonaną inicjalizacją kolejnymi wartościami całkowitymi

```
int[] tablica2 = new int[] { 1, 3, 5, 7, 9 };
```

// Alternatywna składnia

```
int[] tablica3 = { 1, 2, 3, 4, 5, 6 };
```

// Deklaracja dwuwymiarowej tablicy

```
int[,] wielowymiarowaTablica1 = new int[2, 3];
```

//zwróć uwagę na przecinek (,) przy int

// Deklaracja wielowymiarowej tablicy wraz z inicjalizacją wartościami

```
int[,] multiDimensionalArray2 = { { 1, 2, 3 }, { 4, 5, 6 } };
```

Ponadto język C# dopuszcza deklarowanie „tablicy tablic” – nieregularnej tablicy, „postrzępionej” (**jaggedArray**), której elementami są tablice 😊 Mogą one przybierać różne wymiary, rozmiary.

Proszę zapoznać się z: <https://docs.microsoft.com/pl-pl/dotnet/csharp/programming-guide/arrays/jagged-arrays> oraz uruchomić gotowe przykłady.

Aby dostać się do kolejnych elementów tablicy możemy wykorzystać specjalną instrukcję foreach():

a) Przypadek jednowymiarowy

```
C#

int[] numbers = { 4, 5, 6, 1, 2, 3, -2, -1, 0 };
foreach (int i in numbers)
{
    System.Console.Write("{0} ", i);
}
// Output: 4 5 6 1 2 3 -2 -1 0
```

b) przypadek wielowymiarowy

```
C#

int[,] numbers2D = new int[3, 2] { { 9, 99 }, { 3, 33 }, { 5, 55 } };
// Or use the short form:
// int[,] numbers2D = { { 9, 99 }, { 3, 33 }, { 5, 55 } };

foreach (int i in numbers2D)
{
    System.Console.Write("{0} ", i);
}
// Output: 9 99 3 33 5 55
```

Jest to jeden z wielu sposobów – pętli. Najłatwiej użyć pętli for(), jednak w przypadku jej wykorzystania nieodzownie należy ustalić warunki pętli (inicjalizacja zmiennej iterującej, warunek logiczny, instrukcja kroku)

```

static void Main(string[] args)
{
    int[] uczestnicy = new int[] { 19, 34, 23, 54, 31 };
    int[] odwrotnie = new int[uczestnicy.Length];

    // Wpisywanie elementów do tablicy odwrotnie
    for (int i = uczestnicy.Length - 1; i >= 0; i--)
        odwrotnie[uczestnicy.Length - i - 1] = uczestnicy[i];

    // Wyświetlenie elementów tablicy odwrotnie
    for (int i = 0; i < odwrotnie.Length; i++)
        Console.WriteLine(odwrotnie[i]);
    Console.ReadKey();
}

```

//Komentarz

```

static void Main(string[] args)
{
    int[] uczestnicy = { 19, 34, 23, 54, 31 };
    int suma = 0;
    double srednia;
    Console.Write("Wiek uczestników: ");

    for (int i = 0; i < uczestnicy.Length; i++)
    {
        Console.Write("{0}, ", uczestnicy[i]);
        suma += uczestnicy[i];
    }

    srednia = (double)suma / uczestnicy.Length;

    Console.WriteLine();
    Console.WriteLine("Średnia: {0}", srednia );
    Console.ReadKey();
}

```

## 1.6 Instrukcja warunkowa if

1.

```

if (dochod != 0 )
{
    if (dochod > 0)
        Console.WriteLine("Mamy zysk");
    else

```

```

        Console.WriteLine("Mamy stratę");
    }
    else
    {
        Console.WriteLine("Mamy dochód zerowy");
    }
}

```

2.

if – else if – else if (więcej przypadków) - else

### 1.7 Pętla for

```

for ([inicjalizacja]; [wyrażenie logiczne]; [iteracja])
{
    // ciało pętli (instrukcje)
}

```

#### 1.7.1.

```

for (int i = 1 ; i <= 100 ; i++)
for (int i = 0 ; i < 100 ; i++)
for (int i = 20 ; i < 120 ; i++)

for (int i = 1 ; i <= 100 ; i += 2)

for (decimal i = 0.1m; i <= 1; i = i + 0.1m)
//zastosowanie modyfikatora m (decimal) dla literału
    Console.WriteLine(i);

```

## 2. Zadania do samodzielnego rozwiązania

**2.0** Napisz program, który w jednowymiarowej tablicy o rozmiarze 100 umieszcza liczby od 1 do 100, a następnie sumuje liczby parzyste i nieparzyste.

**2.1** Napisz program, który pozwoli zapełnić *n*–elementową tablicę jednowymiarową liczb rzeczywistych wartościami podanymi przez użytkownika. Na początku działania programu użytkownik podaje liczbę elementów tablicy, a następnie poszczególne wartości jej elementów. Po wypełnieniu całej tablicy program powinien wypisać je w oknie konsoli

**2.2** Napisz program **kopiujący** z danej tablicy liczb całkowitych *tab1* do nowej tablicy *tab2* wyłącznie wartości dodatnie. Obie tablice mają być jednowymiarowe o rozmiarze równym 10 (czyli 10-elementowe). Elementy pierwszej tablicy (*tab1*) należy wpisać w trakcie deklaracji tej tablicy.

**2.3** Napisz program, który deklaruje i inicjalizuje **dwuwymiarową tablicę liczb**

**rzeczywistych o rozmiarze 5 x 5.** Program ma wyświetlić elementy tablicy (wiersz po wierszu), a następnie wyświetlić sumę elementów znajdujących się na głównej przekątnej tablicy (główna przekątna – od elementu o indeksach 0,0 do elementu o indeksach  $n,n$ ).

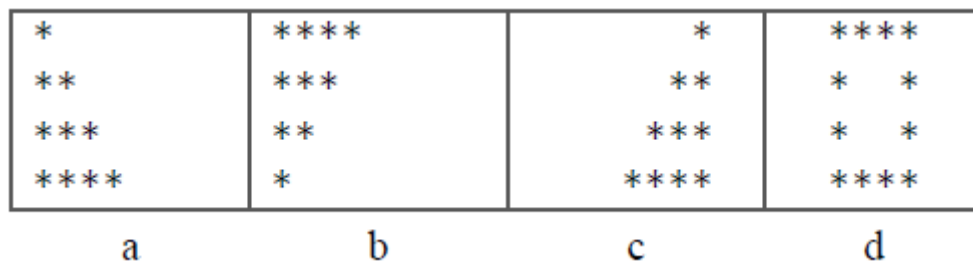
**2.4** Napisz program, który w jednowymiarowej tablicy o rozmiarze 100 umieszcza liczby od 1 do 100, a następnie sumuje liczby parzyste i nieparzyste.

**2.5** Napisz program, który oblicza **wskaźnik masy ciała BMI**. Program ma prosić użytkownika o podanie wagi w kg oraz wzrostu w metrach.

Wzór:  $BMI = (masa/wzrost)$ .

*Wartości na wyjście:* niedowaga ( $< 18,5$ ), wartość prawidłowa ( $18,5-24,99$ ), nadwaga ( $\geq 25,0$ )

**2.6** Napisz program, którego działanie (cztery warianty: a, b, c i d) będzie odpowiadać figurom zbudowanym ze znaku gwiazdki (\*):



Proszę wykorzystać pętlę for().

**2.7** Przeanalizuj wybrane przykłady wykorzystujące metody klasy Array. Spróbuj zmodyfikować swoje programy (z poprzednich laboratoriów/własne przykłady/...) wykorzystując gotowe metody.

**2.8** Napisz program obliczający **liczbę wyrazów** w łańcuchu znaków wprowadzonym przez użytkownika. Należy przyjąć, że wyrazy to ciągi znaków rozdzielone spacją