Nauka C#

Krystian Gronkowski

23 stycznia 2022

Spis treści

1 Hello, world!

1.1 Konfigurowanie kompilatora

Zanim zaczniemy programować musimy zaintalować kompilator, aby być w stanie otworzyć program który napiszemy.

W Linuxie kompilator mcs można zainstalować za pomocą komend:

sudo apt-get update
sudo apt-get install mono-mcs

Natomiast w Windowsie należy najpierw zainstalować .NET Framework, a potem dodać ścieżkę instalacji w zmiennej środowiskowej "PATH"

Alternatywnie, jeżeli ktoś nie chce instalować kompilatora, można użyć jednego z wielu edytorów C# online, np: https://www.onlinegdb.com/online_csharp_compiler

1.2 Console.WriteLine i Console.Readline

Nadszedł czas na stworzenie pierwszego programu!
Struktura piku źródłowego C# wygląda następująco:

```
using System;

class NazwaPliku {
  static void Main() {
    //Kod
  }
}
```

Console.WriteLine() jest funkcją wyświetlającą tekst na ekranie (jak printf w języku C), a **Console.ReadLine()** jest używany do wczytywania tekstu z klawiatury (jak scanf).
Przykładowy program wykorzystujący te dwie funkcje aby wyświetlić imie użytkownika na ekranie:

```
using System;

public class HelloWorld
{
    public static void Main(string[] args)
    {
        Console.WriteLine("Jak sie nazywasz?");
        string imie = Console.ReadLine();
        Console.WriteLine ("Witaj, "+imie+"!");
    }
}
```

1.3 Konwersja danych

Należy wziąć pod uwagę, że **Console.ReadLine()** zawsze wczytuje wartość string, gdybyśmy chcieli aby program wczytywał liczbę zamiast imienia (np wiek), musielibyśmy przekonwertować string do wartości int.
Na szczęście jest do tego wbudowana funkcja **Int32.Parse(string)**. Przykład jej użycia:

```
using System;

public class HelloWorld
{
    public static void Main(string[] args)
    {
        Console.WriteLine("Ile masz lat?");
        int wiek = Int32.Parse(Console.ReadLine());
        if(wiek>17){
            Console.WriteLine("Jestes pelnoletni");
        }
        else{
            Console.WriteLine("Nie jestes pelnoletni");
        }
    }
}
```

2 Operowanie na danych

2.1 String.Contains

String.Contains() jest bardzo użyteczną funkcją, która sprawdza czy gdziekolwiek w jakimś łańcuchu można znaleźć inny łańcuch. Zwraca wartość boolowską. Przykładem jego zastosowania jest sprawdzenie czy w jakimś słowie występuje litera 'a' :

```
using System;

class Contains {
    static void Main() {
      Console.WriteLine("Wpisz slowo bez litery a");
        string x = Console.ReadLine();
        if(x.Contains('a')){
          Console.WriteLine("W tym slowie nie ma litery a.");
        }
        else{
          Console.WriteLine("W tym slowie jest litera a.");
        }
    }
}
```

2.2 String.Replace

String.Replace() zastępuje wszystkie wystąpienia łancucha A innym łańcuchem B.

```
using System;

class Replace {
    static void Main() {
        string x = "Ala ma kota.";
        x = x.Replace("kota","psa");
        Console.WriteLine(x);
        // W konsoli zostanie wyswietlone "Ala ma psa."
    }
}
```

2.3 String.Split

String.Split() rozdziela łańcuch na tablicę mniejszych łańcuchów rozdzielonych za pomocą łańcucha podanego podczas wywoływania funkcji. Bardzo użyteczne jeżeli chcemy odczytać dane rozdzielonych za pomocą nowej linijki (") albo spacji. Przykład użycia:

```
using System;

class Split {
    static void Main() {
        Console.WriteLine("Wpisz swoje imie i nazwisko");
        string x = Console.Readline();
        string[] input = x.Split(' ');
        string imie = input[0];
        string nazwisko = input[1];
        Console.WriteLine("Dzien dobry, Pani/e " + nazwisko);
    }
}
```

Zadanie 2.1. Poproś użytkownika o wpisanie jakiegoś zdania, a potem wyświetl je ze słowami w odwrotnej kolejności. Ala ma kota → kota ma Ala

Zadanie 2.2. Poproś użytkownika o wpisanie jakiegoś słowa i wyświetl wszystkie litery które w nim nie występują.

3 Tablice i listy

3.1 Tablice jednowymiarowe

Tablica to zbiór uporządkowanych elementów tego samego typu. Każdy element można wyczytać podając numer jego indeksu. Przykład zastosowania tablicy aby wyczytać ile razy każda cyfra występuje w jakiejś liczbie.

```
using System;

class HelloWorld {
    static void Main() {
      int[] tablica = new int[10];
        string liczba = "1049284883920457162047859300018306";
        for(int i=0;i<liczba.Length;i++){
          tablica[Int32.Parse(liczba[i].ToString())]+=1;
        }
        for(int i=0;i<10;i++){
          Console.WriteLine("Liczba " + i + " wystepuje " + tablica[i] + " razy.");
        }
    }
}
```

3.2 Tablice wielowymiarowe

Najprostszy sposób patrzenia na tablice dwuwymiarową tablica[k][w] jest wyobrażenie sobie tabelki o k kolumnach i w wierszach.
Dla przykładu, wyobraźmy sobie tablicę dwuwymiarową [5][3] wypełnioną w następujący sposób:

1	4	6	2	0
2	7	7	0	3
-15	55	26	0	330

tablica[1][0] nawiązuje do liczby w 2 kolumnie i w 1 wierszu (pamiętaj że tablica zaczynają się od 0!), czyli do liczby 4. W ten sam sposób tablica[2][2] dałaby wynik 26, a tablica[0][0] 1.

Zadanie 3.1. Stwórz tablicę dwuwymiarową[10][10] i wypełnij ją tabliczką mnożenia.

3.3 Listy

Listy działają jak tablice, ale nie mają zadeklarowanej wielkości. Można dodawać nowe elementy do listy i usuwać stare gdy tylko się chce. Jest to bardzo użyteczne, gdy chcemy przechować jakieś elementy, ale nie wiemy jak jest ich dużo. Na przykład gdy chcemy wczytywać liczby podawane przez użytkownika, dopóki nie wpisze 0.

```
using System;
using System.Collections.Generic; //Biblioteka w ktorej znajduja si Ź listy

class HelloWorld {
    static void Main() {
        List<int> ints = new List<int>();
        while(true){
            int x = Int32.Parse(Console.ReadLine());
            if(x!=0){
                ints.Add(x);
            }
            else{
                break;
            }
        }
        foreach(int x in ints){
            Console.WriteLine(x);
        }
    }
}
```

4 Losowanie liczb

Literatura

[1] https://automatykanacodzien.pl/2020/07/24/kompilacja-i-uruchomienie-kodu-c-w-linii-polecen-windows-linux/