
Tema 1

- Snalazenje u Visual Studio okruzenju
- Kreiranje praznog projekta - **console app(.NET core)**
- Sintaksna pravila klasa i namespacea
- Osnovni tipovi podataka u C#
- Aritmeticke operacije nad njima
- Kastovanje podataka
- WriteLine(), Readline()

Namespaces:

- ❖ Pandan sa Pythonovim importima - razlika je u tome sto je C# vise oop nego python pa se kroz jedan namespace mogu da organizuju razne klase koje se medjusobno nasleduju
- ❖ System biblioteka (ili c-sharpovski -> namespace, a matematički kao oblast definisanosti) sadrzi klasu Console, gde npr imamo funkcije za ispis u upis podataka kroz konzolu do programa. Koriscenjem kljucne reci **using** mozemo da direktno pozivamo funkcije WriteLine() i ReadLine()

PRIMER 1 - namespaces:

Napisati razlicite namespaceove i istestirati ko koga vidi.

POSTUPAK:

```
using System;
```

```
namespace drugiNamespace
{
    public class Pozdrav
    {
        public void sayHello() { Console.WriteLine("Pozdrav iz drugog namespacea"); }
    }
}
```

```
namespace TreciNamespace
{
    public class Cao
    {
        public void sayHello() { Console.WriteLine("Pozdrav iz treceg namespacea!"); }
    }
}
```

```
namespace Cas01_Primer01_namespaces
{
    class Program
```

```

{
    static void Main(string[] args)
    {
        Console.WriteLine("Prvi namespace!");
        drugiNamespace.Pozdrav pz = new drugiNamespace.Pozdrav();
        TreciNamespace.Cao cao = new TreciNamespace.Cao();
        pz.sayHello();
        cao.sayHello();
    }
}

```

OBJASNJENJE RESENJA:

NAPOMENA 1: Uraditi narednu verziju koristenjem usinga:

```

using System;
using drugiNamespace;
using TreciNamespace;

namespace drugiNamespace
{
    ...
}

namespace TreciNamespace
{
    ...
}

namespace Cas01_Primer01_namespaces
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Prvi namespace!");
            Pozdrav pz = new drugiNamespace.Pozdrav();
            Cao cao = new TreciNamespace.Cao();
            pz.sayHello();
            cao.sayHello();
        }
    }
}

```

PRIMER 2 - osnovni tipovi podataka

V1 - zbir 3 integera:

```

static void Main(string[] args)
{
    int a, b, c, x;
    string prvi, drugi, treci, prosek;
    Console.WriteLine("Unesite 3 broja, pa da izracunam njihov
zbir:");
}

```

```

    prvi = Console.ReadLine();
    drugi = Console.ReadLine();
    treci = Console.ReadLine();
    a = Convert.ToInt32(prvi); // probati bez konverta, da se vidi!
    b = Convert.ToInt32(drugi);
    c = Convert.ToInt32(treci);
    x = (a + b + c);
    prosek = Convert.ToString(x);
    Console.WriteLine("Zbir iznosi:" + prosek);
    Console.ReadKey();
}

```

V2 - avg od ta tri broja

```

static void Main(string[] args)
{
    int a, b, c;
    double x;
    string prvi, drugi, treci, prosek;
    Console.WriteLine("Unesite 3 broja, pa da izracunam njihov
zbir:");

    prvi = Console.ReadLine();
    drugi = Console.ReadLine();
    treci = Console.ReadLine();
    a = Convert.ToInt32(prvi);
    b = Convert.ToInt32(drugi);
    c = Convert.ToInt32(treci);
    x = (double)(a + b + c)/3;
    prosek = Convert.ToString(x);
    Console.WriteLine("Zbir iznosi:" + prosek);
    Console.ReadKey();
}

```

V2 - unos moze biti double

```

static void Main(string[] args)
{
    double a, b, c;
    double x;
    string prvi, drugi, treci, prosek;
    Console.WriteLine("Unesite 3 broja, pa da izracunam njihov
zbir:");

    prvi = Console.ReadLine();
    drugi = Console.ReadLine();
    treci = Console.ReadLine();
    a = Convert.ToDouble(prvi);
    b = Convert.ToDouble(drugi);
    c = Convert.ToDouble(treci);
    x = (double)(a + b + c)/3;
    prosek = Convert.ToString(x);
}

```

```

        Console.WriteLine("Zbir iznosi:" + prosek);
        Console.ReadKey();
    }

```

PRIMER 3 - TIPOVI PODATAKA

int	4 bajta	Brojevi izmedju -2147483648 i 2147483647
long	8 bajtova	-9,223,372,036,854,775,808 i 9,223,372,036,854,775,807
float	4 bajta	7 decimalnih mesta
double	8 bajtova	15 decimalnih mesta
bool	1 bit	Tacno ili netacno vrednost
char	2 bajta	Pod jednostrukim navodnicima jedan karakter
string	2 bajta*len	Pod dvostrukim navodnicima niz karaktera

POSTUPAK:

```

static void Main(string[] args)
{
    int myNum = 5; // Integer (whole number)
    double myDoubleNum = 5.99D; // Floating point number
    char myLetter = 'D'; // Character
    bool myBool = true; // Boolean
    string myText = "Hello"; // String
    long myLongNum = 15000000000L;
    float myFloat = 35e5F;

    Console.WriteLine(myNum);
    Console.WriteLine(myDoubleNum);
    Console.WriteLine(myLetter);
    Console.WriteLine(myBool);
    Console.WriteLine(myText);
    Console.WriteLine(myLongNum);
    Console.WriteLine(myFloat);
}

```

OBJASNJENJE RESENJA:

NAPOMENA 1: Ispitati da li su jasna predstavljanja podatka u obliku bitova i bajtova

PRIMER 4 - BROJ 8

```

static void Main(string[] args)
{
    Console.WriteLine("Unesi jedan broj: ");
    int broj = Convert.ToInt32(Console.ReadLine());
    int rezultat = broj;

    Console.WriteLine("Oduzmi broj 1 od njega... ");
}

```

```

rezultat = rezultat - 1;
Console.ReadKey();

Console.WriteLine("Pomnozi rezultat sa brojem 3... ");
rezultat = rezultat * 3;
Console.ReadKey();

Console.WriteLine("Dodaj broj 12 na rezultat... ");
rezultat = rezultat + 12;
Console.ReadKey();

Console.WriteLine("Podeli broj sa 3... ");
rezultat = rezultat / 3;
Console.ReadKey();

Console.WriteLine("Dodaj broj 5... ");
rezultat = rezultat + 5;
Console.ReadKey();

Console.WriteLine("Oduzmi zamisljeni broj... ");
rezultat = rezultat - broj;
Console.ReadKey();

Console.WriteLine("Dobio si broj 8?");
Console.WriteLine(rezultat);
Console.ReadKey();
}

```

OBJASNJENJE RESENJA:

NAPOMENA 1: Pokazati operatore +=, -=, ..

PRIMER 5 - OBJEKTNO ORJENTISANI NACIN RAZMISLJANJA

- Proceduralno vs objektno orjentisano programiranje
- Proceduralno - postoje globalni podaci koje svako vidi i svako koristi. Ne prati se da neko i moze da ih promeni (npr baza podataka, ili prenos paketa podataka preko mreze od servera do klijenta gde se prenose samo podaci, a sam server i klijent zna kako da cita taj podatak)
- U oop smislu, prenosio bi se ceo objekat, i podaci i ponasanje tog objekta - podaci su atributi objekta, ako imamo objekat sluzbenik, prenosili bi se njegovi atributi, ime, prezime, jmbg,.. Ponasanje objekta je sta objekat moze da radi - opisuje se metodima koje se pozivaju slanjem odredjene poruke - get jmbg, ,set jmbg. Za svaku metodu je bitno da se zna - ime, parametri koji se prosledjuju i povratni tip.
- Uml klasni dijagrami
- Pravljenje primera objekta iz klase. Klase su kao sabloni modelovanja, sematski prikaz tj plan jednog objekta
- Klase i objekti - objasnjenje je kao kokoska i jaje, jedno bez drugog ne moze da se objasni. Jedino sto je u programiranju jasno jeste da se uvijek - prvo kreira klasa pa tek onda objekti.

