

# Vježba 1 - Uvod u C# i Visual Studio

## Uvod

Na ovim vježbama će se koristiti razvojno okruženje .NET i alat Visual Studio. Prva vježba prolazi kroz osnove programskog jezika C#. U narednim vježbama se od studenta očekuje da je upoznat s osnovama C# i s osnovnim konceptima objektnog programiranja (klase, nasljeđivanje...)

## Razvojno okruženje Visual Studio i C#

📺 Visual Studio C# Tutorial for Beginners EP01 - First Steps Hello World Console App

### ZADATAK 1:

Provjeriti verziju aplikacije Visual Studio i .NET arhitekture i dohvatiti najnoviju verziju koda iz git repozitorija



### ZADATAK 2:

Izraditi novu konzolnu aplikaciju u Visual Studiu.

Create new project - Installed - Visual C# - Windows Desktop - Console App - Nazvati TppImePrezime1

- Console App (.NET Framework)

Osnovna struktura aplikacije:

- Solution - skup projekata koji se pokreću zajedno
- Project - sadrži sve datoteke koje se kompajliraju u izvršnu datoteku.
- Klasa - Osnovna jedinica koda
- Datoteka Program.cs - početna klasa koja se automatski izvršava
- Metoda main - metoda koja se automatski pokreće kod izvršavanja aplikacije

### ZADATAK 3:

U konzolnu aplikaciju implementirati osnovne koncepte C# prema uputama

## Hello World

1	U klasi Program, metodi Main, napisati poruku "Hello World" i pokrenuti aplikaciju (ctrl + F5)
<pre>namespace Vjezba1 {     class Program     {         static void Main(string[] args)         {             Console.WriteLine("Hello World");         }     } }</pre>	

## Varijable i konstante

2	Deklarirati varijablu broj tipa integer i konstantu, pokušati ih izmijeniti
<pre>namespace Vjezba1 {     class Program     {         static void Main(string[] args)         {             int variable = 5;             const float Constant = 3.14f;              variable = 6;             Constant = 4;         }     } }</pre>	

## Formatiranje varijabli za ispis

3	Na konzolu u poruci ispisati numeričku i tekstualnu vrijednost
<pre>namespace Vjezba1 {     class Program     {         static void Main(string[] args)         {             int variable = 5;             const float Constant = 3.14f;              Console.WriteLine("Variable:{0}, constant:{1}", variable, Constant);         }     } }</pre>	

## Konverzija tipova: implicitna i eksplicitna, pomoću .NET metoda.

4	Implicitno pretvoriti vrijednost int u long
<pre>static void Main(string[] args) {     int integerValue = 5;     long longIntegerVariable = 10;     string stringVariable = "a";      longIntegerVariable = integerValue;     integerValue = longIntegerVariable;     stringVariable = integerValue; }</pre>	
<p>NAPOMENA: implicitno se mogu pretvoriti samo tipovi koji su kompatibilni, tako da tip u kojeg se pretvara zauzima isto ili više memorije od tipa iz kojeg se pretvara. (int se može pretvoriti u long, ali long ne može u int)</p>	

5	Eksplicitno pretvoriti vrijednost tipa int u byte
<pre>static void Main(string[] args) {     int integerValue = 257;     byte byteVariable = 6;      byteVariable = (byte)integerVariable;     Console.WriteLine("byteVariable:{0}", byteVariable); }</pre>	

## Operatori

6	Aritmetički operatori
<pre>static void Main(string[] args) {     float a = 3;     float b = 4;     Console.WriteLine("a + b: {0} + {1} = {2}", a, b, a + b);     Console.WriteLine("a - b: {0} - {1} = {2}", a, b, a - b);     Console.WriteLine("a * b: {0} * {1} = {2}", a, b, a * b);     Console.WriteLine("a / b: {0} / {1} = {2}", a, b, a / b);     Console.WriteLine("a % b: {0} % {1} = {2}", a, b, a % b); }</pre>	
7	Operatori za uspoređivanje
<pre>static void Main(string[] args) {     float a = 3;     float b = 4;     Console.WriteLine("a == b: {0} == {1} = {2}", a, b, a == b);     Console.WriteLine("a != b: {0} != {1} = {2}", a, b, a != b);     Console.WriteLine("a &gt; b: {0} &gt; {1} = {2}", a, b, a &gt; b);     Console.WriteLine("a &gt;= b: {0} &gt;= {1} = {2}", a, b, a &gt;= b);     Console.WriteLine("a &lt; b: {0} &lt; {1} = {2}", a, b, a &lt; b);     Console.WriteLine("a &lt;= b: {0} &lt;= {1} = {2}", a, b, a &lt;= b); }</pre>	
8	Logički operatori
<pre>static void Main(string[] args) {     bool a = true;     bool b = false;     Console.WriteLine("a &amp;&amp; b: {0} &amp;&amp; {1} = {2}", a, b, a &amp;&amp; b);     Console.WriteLine("a    b: {0}    {1} = {2}", a, b, a    b);     Console.WriteLine("!a:      !{0} = {1}", a, !a); }</pre>	
9	Bitwise operatori
<pre>static void Main(string[] args) {     int a = 0b10;     int b = 0b100;     Console.WriteLine("a &amp; b: {0} &amp; {1} = {2}", a, b, a &amp; b);     Console.WriteLine("a   b: {0}   {1} = {2}", a, b, a   b); }</pre>	

## Polja

10	Polja
<pre>static void Main(string[] args) {     string[] words = new string[4];      words[0] = "Hello";     words[1] = ",";     words[2] = "world";      Console.WriteLine("First Word: {0}", words[0]); }</pre>	

## Enumeratori

11	Enum
<pre>enum Statusi { Redovni = 1, Izvanredni = 2 }  static void Main(string[] args) {     Statusi prviStatus = Statusi.Redovni;     Console.WriteLine("Prvi Status: {0}", prviStatus);      string nazivStatusa = "Izvanredni";      var drugiStatus = (Statusi)Enum.Parse(typeof(Statusi), nazivStatusa);     Console.WriteLine("Drugi Status: {0}", drugiStatus);      var treciStatus = 1;     Console.WriteLine("Treći Status: {0}", (Statusi)treciStatus); }</pre>	

## Grananje

12	IF naredba
<pre>static void Main(string[] args) {     var temp = 20;      if (temp &lt; 0)     {         Console.WriteLine("Temperatura je: {0}, Hladno", temp);     }     else if (temp &lt; 20)     {         Console.WriteLine("Temperatura je: {0}, Ugodno", temp);     }     else     {         Console.WriteLine("Temperatura je: {0}, Vruće", temp);     } }</pre>	

13	uvjetni operator
<pre>static void Main(string[] args) {     var temp = 20;</pre>	

```
// uvjetni operator ? :
var opisno = temp < 0 ? "Hladno" : temp < 20 ? "Ugodno" : "Vruće";
Console.WriteLine("Temperatura je {0}", opisno);

}
```

## 14 switch - case

```
static void Main(string[] args)
{
    var godina = 1;
    switch (godina)
    {
        case 1:
            Console.WriteLine("Prva godina");
            break;
        case 2:
            Console.WriteLine("Druga godina");
            break;
        default:
            break;
    }
}
```

## Klase

### 15 Deklaracija klase

```
public class Osoba
{
    public string Ime;
    public Osoba(string ime)
    {
        this.Ime = ime;
    }
    public string KakoSeZove()
    {
        return this.Ime;
    }
}
```

### 16 Pozivanje klase

```
Osoba Ivo = new Osoba("Ivo");
Console.WriteLine( Ivo.KakoSeZove() );
```

### 17 Interface

```
public interface IOsoba
{
    string KakoSeZove();
}
```

### 18 Implementacija interface-a

```
public class Osoba : IOsoba
{
    ...
}
```

19	Nasljeđivanje
<pre> public class Programer: Osoba {     public int GodineIskustva;      public Programer(string ime, int godineIskustva):base(ime)     {         this.GodineIskustva = godineIskustva;     } }  Programer Ivo = new Programer("Ivo", 4); Console.WriteLine(\$"Programer {Ivo.KakoSeZove()} ima {Ivo.GodineIskustva} godina iskustva."); </pre>	

## Konzola

20	Čitanje ulaza
<pre> string ulaz = Console.ReadLine(); </pre>	

### ZADATAK 4:

Izraditi konzolnu aplikaciju koja će na ulazu primiti ime djelatnika i njegovu plaću. Na temelju ulaza će instancirati klasu Djelatnik, koja nasljeđuje već postojeću klasu Osoba.

Ako djelatnik ima plaću veću od 8000 kn, program će javiti poruku "Djelatnik ima veliku plaću". U suprotnom će javiti poruku "Djelatnik ima malu plaću".

Aplikacija mora provjeriti je li unos plaće numerički i je li u rasponu od 0 do 100000 i ako nije mora javiti grešku: "Unos mora biti između 0 i 100000"