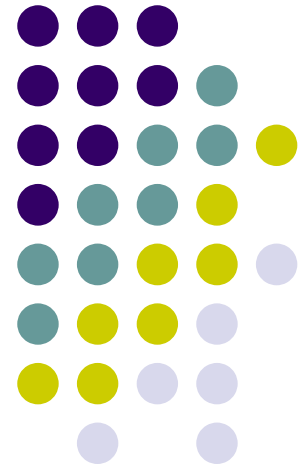


Računarski praktikum 3

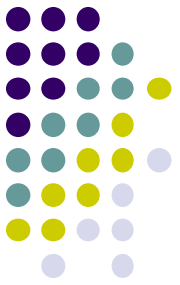
Vježbe 01

Vinko Petričević

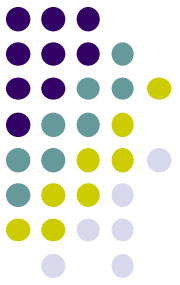
**Uvod, osnove jezika,
osnovni tipovi, metode**



Gradivo i način polaganja

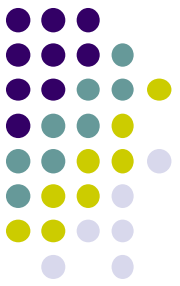


- Gradivo:
 - Osnove jezika C# i platforme .NET
 - Windows Forms



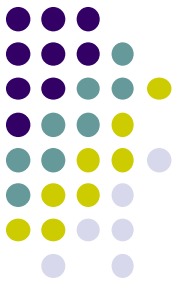
Osnove jezika

- C# je dio .NET platforme
- .NET platformu čine:
 - Zajednički sustav tipova (CTS);
 - Zajednička standardna biblioteka (CoreFX, Framework Class Library (FCL));
 - Zajednička virtualna mašina (CoreCLR, CLR)
- Standard CLI, međujezik CIL
- .NET Framework, .NET Core



Osnove jezika

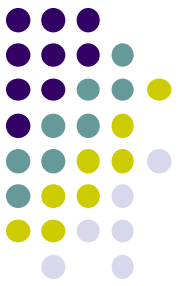
- Instance klasa alocirane su na heapu (dinamički), a o dealokaciji brine garbage collector
- Struktura je jednostavniji tip. Zauzima prostor na stogu. Struktura se ne može naslijediti iz klase, niti ju se može naslijediti, ali može implementirati sučelje
- Klasa se može naslijediti samo iz jedne bazne klase, ali može implementirati više sučelja
- Podrška za delegate, verziju pokazivača na funkcije



Osnove jezika

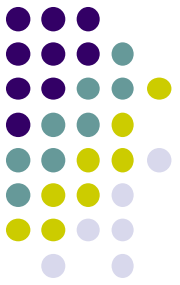
- U Visual Studiu kod se organizira u projekte
- Svaki projekt sadrži neke obavezne datoteke, ali možemo dodati proizvoljan broj datoteka i poddirektorija (konvencija je svaku klasu pisati u zasebnoj datoteci)
- Kompajliranjem iz svih datoteka projekta nastaje jedan assembly (.dll ili .exe).

Jednostavan program



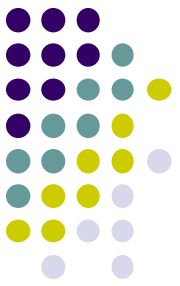
```
class Hello
{
    static void Main()
    {
        // ispis teksta na ekran
        System.Console.WriteLine("Hello World");
    }
}
```

- Visual Studio: File -> New -> Project...
 - Project Types: C#, Windows
 - Templates: Console Application
 - Biramo najnoviju .NET platformu, **bez** „Framework“
 - Ne koristimo top-level naredbe (zasad)
 - Odabрати ime (npr: Hello)
- *dotnet build* i *csc.exe*



Zadatak 1

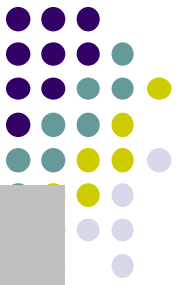
- Koristeći Visual Studio, napišite program koji ispisuje pozdravnu poruku
- Uočite razlike u programu kojeg je definirao wizard



Osnove jezika

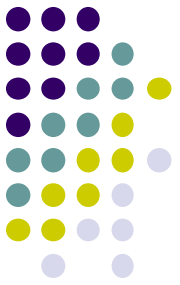
- C# je strogo tipizirani jezik
- Imamo ugrađene i korisnički definirane tipove
- Prema načinu spremanja u memoriji dijele se na vrijednosne (stog) i referentne (heap, GC)

Vrijednostni tipovi



Tip	Veličina (u bajtovima)	.NET tip	Opis
bool	1	Boolean	<i>true</i> ili <i>false</i>
byte	1	Byte	od 0 do 255 (od 0 do 2^8-1)
char	2	Char	pojedinačna <i>UTF-16</i> kodna jedinica
sbyte	1	SByte	od -128 do 127 (od -2^7 do 2^7-1)
short	2	Int16	od -32 768 do 32 767 (od -2^{15} do $2^{15}-1$)
ushort	2	UInt16	od 0 do 65 535 (od 0 do $2^{16}-1$)
int	4	Int32	od -2 147 483 648 do 2 147 483 647 (od -2^{31} do $2^{31}-1$)
uint	4	UInt32	od 0 do 4 294 967 295 (od 0 do $2^{32}-1$)
float	4	Single	7 decimala, od $\pm 1,5 \times 10^{-45}$ do $\pm 3,4 \times 10^{38}$
double	8	Double	15-16 decimala, od $\pm 5,0 \times 10^{-324}$ do $\pm 3,4 \times 10^{308}$
decimal	16	Decimal	28-29 decimala, od $\pm 1,0 \times 10^{-28}$ do $\pm 3,4 \times 10^{28}$
long	8	Int64	od -9 223 372 036 854 775 808 do 9 223 372 036 854 775 807 (od -2^{63} do $2^{63}-1$)
ulong	8	UInt64	od 0 do 18 446 744 073 709 551 615 (od 0 do $2^{64}-1$)

Enum i struct



Osnove jezika

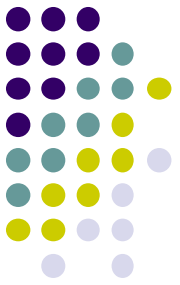
- Implicitne pretvorbe rade, ako se ne gubi preciznost:

```
int x = 5;  
double y = x;
```

- Za obrnuti smjer moramo biti eksplicitni:

```
double x = 5;  
int y = (int)x;
```

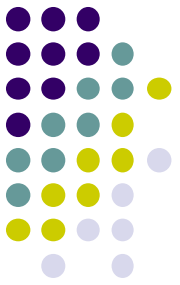
Jednostavan program

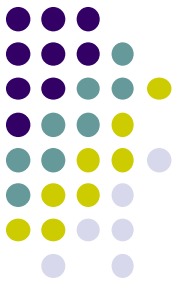


```
namespace ConsoleApplication1
{
    class Program
    {
        static void Main()
        {
            string s = Console.ReadLine();
            int x = Convert.ToInt32(s);
            for (int i = 0; i < x; ++i)
                Console.WriteLine(i);
        }
    }
}
```

Jednostavan program

```
int x;  
try  
{  
    x = Convert.ToInt32(s);  
}  
catch  
{  
    x = 10;  
}
```





Varijable

- Prije korištenja, svaku varijablu moramo inicijalizirati

```
int x = 5, y;  
y = 10; // ako zakomentiramo, komp. greška  
Console.WriteLine("x = {0}, y = {1}", x, y);
```

- Varijablama možemo mijenjati vrijednost
- Konstante moramo odmah inicijalizirati, i kasnije im se vrijednost ne može mijenjati

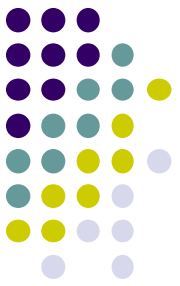
```
const int z = 15;
```

string



- string je niz znakova

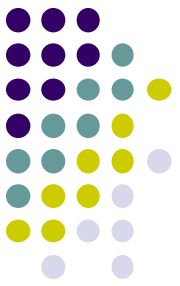
```
string s = "abc def";  
s = "abc\ndef"; // \n se tretira kao novi red  
s = @"abc\ndef"; // ispisuju se znakovi \ i n  
s = Console.ReadLine();
```



Operatori i tijek programa

- Operatori su slični kao u jeziku C++
- Kontrola toka vrlo slična:
 - foreach – prolazi kroz sve elemente nekog spremnika
 - switch – ne mora biti cjelobrojni tip i možemo koristiti goto case

```
switch(i) {  
    case 1: Console.Write(1); goto case 3;  
    case 2: Console.Write(2); break;  
    case 3: Console.Write(3); break;  
    default: Console.Write("nesto drugo"); break;  
}
```



Slanje parametara funkciji

```
static void f1(int x) { ++x; }
static void f2(ref int x) { ++x; }
static void f3(out int x) {
    // ++x; ovo bi bila greska
    x = 0; ++x;
}
static void Main(string[] args)
{
    int x = 10;
    f1(x); Console.WriteLine(x); // 10
    f2(ref x); Console.WriteLine(x); // 11
    f3(out x); Console.WriteLine(x); // 1
}
```

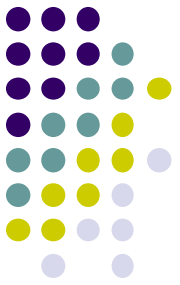



Enumeracije

```
enum imeEnumeracije [:osnovniTip = int]
{
    const1 = vrijednost1, // ako ne pridružimo vr., počinje od nule
    const2 = vrijednost2, // ako ne pridružimo vr., inkrementira se
    // prethodna vrijednost
    ...
}
```

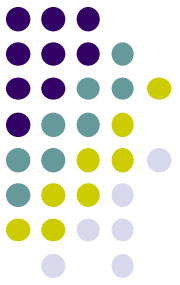
```
enum Temperature {
    ledisteVode = 0,
    vrelisteVode = 100,
}

static void Main(string[] args) {
    Console.WriteLine("Ledište vode: {0}",
        (int)Temperature.ledisteVode);
    Console.WriteLine("Vrelište vode: {0}",
        (int)Temperature.vrelisteVode);
}
```



Zadatak 2

- Napišite program koji za uneseni broj provjerava je li prost
- Napišite program koji uneseni broj rastavlja na proste faktore



Zadatak 3

- Napišite funkciju koja vraća najmanji prosti broj veći od danog broja
- Napišite funkciju koja ispisuje sve proste brojeve manje od danog broja
- Napišite funkciju koja od dobivenog stringa napravi palindrom