Uvod u programiranje kroz programski jezik C#

Skripta za kurs: “C# Osnovni”

Lazar Premović
avgust 2019.

Uvod

Ova skripta je sačinjena iz dva glavne celine: objašnjenja i podsetnici za teme obrađene tokom kursa i zbirka zadataka sa rešenjma koja pokriva sve teme obrađene na kursu i može se koristiti kako za rad na casu tako i za samostalan rad.

Uvod u C# Osnovni kurs

Kurs “C# Osnovni” je namenjen učenicima starijih razreda osnovnih škola i učenicima srednjih škola koji do sada nisu imali prilike da se upoznaju sa procesom programiranja i kao takav ne zahteva nikakvo predznanje iako je poznavanje Engleskog jezika poželjno.

Kroz ovaj kurs moći ćete da se upoznate sa osnovama procesa programiranja na primerima u programskom jeziku C#, za početak ćemo raditi na konzolnim aplikacijama dok se ne upoznamo sa konceptima grananja i petlji a onda ćemo se pozabaviti i aplikacijama sa grafičkim interfejsom.

Programski jezik C#

C# je programski jezik opšte primene koji je Microsoft razvio 2000. godine za potrebe svoje .NET platforme. C# pripada familiji C jezika, bazira se na imperativnoj i proceduralnoj paradigmi i podžava objektno orijentisano programiranje.

C# je izabran kao jezik koji ćemo koristiti na ovom kursu zato što je veoma korišćen u industriji za izradu ozbiljnih aplikacija a istovremeno je i dovojno jednostavan za početnike i omogućava vrlo lako kreiranje čak i aplikacija sa grafičkim interfejsom.

Preuzimanje Microsoft Visual Studio programerskog okruženja

Da bi programirali u programskom jeziku C# potrebno je da koristimo programersko okruženje Microsoft Visual Studio koje je potrebno da instaliramo na sledeći način:

Na sajtu https://visualstudio.microsoft.com/ možete preuzeti Microsoft Visual Studio (vodite računa da preuzmete verziju Community 2019), ili koristite direktni link https://visualstudio.microsoft.com/thank-you-downloading-visual-studio/?sku=Community.
Pri instalaciji vodite računa da čekirate podršku za C# jezik.
Nakon što instalirate Visual Studio potrebno je da obavite početnu konfiguraciju, biranje šeme boja (nije preterano bitno) i kreiranje Microsoft naloga sa kojim je neophodno da se ulogujete da biste koristili softwer.

Kako se koristi programersko okruženje će biti objašnjeno u osnovnim crtama i nećemo se time baviti u ovoj skripti.

0. Uvod u proces programiranja i izvršavanja koda

Kada programiramo bilo koju aplikaciju postoje dve najbitnije faze u njenoj izradi, to su: faza razade algoritma i faza kodiranja.

Algoritmi i faza razrade algoritama

Algoritam je drugi naziv za konačan niz jednostavnih koraka kojima se opisuje postupak rešavanja određenog realnog problema. A to znači da u fazi razrade algoritma mi uzimamo postavku problema (U našem slučaju tekst zadatka) i predstavljamo proces za njegovo rešavanje kao niz jednostavnih koraka koje računar može da razume (tj. Algoritam).
Ova faza će od vas zahtevati da pokažete sposobnosti logičkog razmišljanja.

Neki primeri algoritama u svakodnevnom životu:

Kuvanje jajeta:

1. Uzmemo lonče i napunimo ga vodom,
2. Uzmemo jaje i stavimo ga u lonče,
3. Stavimo lonče na ringlu i uključimo tu ringlu,
4. Čekamo da voda proklјuča,
5. Čekamo 5 minuta da se jaje skuva,
6. Sklonimo lonče sa ringle i isključimo tu ringlu,
7. Sačekamo da se jaje ohladi ili ga ohladimo hladnom vodom, kraj postupka.

Telefoniranje:

1. Podižemo slušalicu,
2. Ako nemamo odgovarajući signal, kraj postupka (nesupešno telefoniranje),
3. Biramo broj,
4. Ako se pozvani korisnik javi, razgovaramo
5. Prekidamo vezu, kraj postupka.

Programski kod i faza kodiranja

Kada imamo razrađen algoritam koji smo napisali na normalnom jeziku, potrebno ga je prevesti na niz komandi programskog jezika koji smo odabrali, to je faza kodiranja zato što se niz komandi na programskom jeziku naziva kod. Programski kod koji pišemo na nekom programskom jeziku ima neka svoja pravila isto kao što i normalni jezici imaju svoju gramatiku.

Kodiranje u programskom jeziku C#

C# je jezik u kome se komande i programski kod pišu u vidu teksta i za njih koristi standardnu englesku abecedu (mala i velika skola, cifre i specijale znake).
Svaka komanda u C#-u se završava tačka zarezom ; i dobra praksa je pisati po jednu komandu u jednom redu jer je tako lakše videti redosled izvršavanja komandi.
Komande u C#-u se izvršavaju odozgo na dole (i sa leva na desno ako ima više komandi u jednom redu). Kada pokrenemo program on se izvršava od prve linije koda redom naniže do poslednje linije koda, i kada izvrši poslednju liniju koda program prestaje sa radom i gasi se.
U skoro svakom programskom jeziku postoje komentari, u C# se pišu sa dve kose crte `//` računar će ingorisati bilo šta nakon njih što nam omogućava da u njima mi zapišemo neke naše beleške

```
int a = 5;
int b = 3;
int c = a + b;
//komentar
```

Primer nekoliko linija koda

1. Osnovni programi linijske strukture

Specificnosti izvršavanja konzolnih aplikacija

Jedina specificnost kod konzolnih apllikacija u ondosu na način izvršavanja o kome smo do sada pričali je da će programersko okruženje pri kreiranju projekta konzolne aplikacije nama kreirati šablon u koji je potrebno da ubacimo naš kod. U ovom slučaju, programersko okruženje će kreirati funkciju `Main` unutar koje mi pišemo naš kod.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            //Nas kod pisemo ovde
        }
```

```
    }
}
```

Primer šablona za konzolne aplikacije

Varijable i osnovni tipovi podataka

Jedan od osnovnih koncepata u programiranju su varijable, varijable su način na koji možemo da obrađujemo i čuvamo podatke unutar programa. Varijable možemo najlakše zamisliti kao kutije koje na sebi imaju obeleženo ime i koju vrstu podataka čuvaju i u njih možemo staviti neku vrednost i posle je uzeti iz kutije i koristiti negde drugde.

Malopre smo spomenuli da svaka varijabla ima: ime, tip podataka koji sadrži i vrednost koju čuva, u C#-u da bi mogli da koristimo neku varijablu moramo najpre da je deklarišemo. Pri deklaraciji obaveštavamo računar koje je ime varijable koju ćemo koristi i njen tip.

```
<Tip varijable> <Ime varijable>;

int broj;
string tekst;
float josJedanBroj;
```

Primeri deklaracija varijabli

int	Celobrojna vrednost
float	Decimalni broj
char	Jedan karakter (slovo, cifra, specijalni karakter)
string	Tekst (sačinjen od proizvoljnog broja karaktera)
bool	Logička vrednost (tačno **true** ili netačno **false**)

Osnovni tipovi podataka

Pravila za imenovanje varijabli:

- Mogu da se sastoje samo od slova (malih i velikih), cifara i donjih crta (_)
- Ne smeju počinjati cifrom
- C# razlikuje mala i velika slova, `GodisnjeDoba` i `godisnjeDoba` nisu ista varijabla
- Varijable se ne smeju zvati po ključnim rečima C# jezika (neke od klučnih reči su: `do,for,if,class`)

Naredba dodele i osnovni numerički operatori

Osnovna naredba u C#-u je naredba dodele, ona nam ovogućava da neku vrednost (dobijenu kao rezultat nekog izraza) zabeležimo u nekoj varijabli (**Napomena**: varijabla mora pre toga biti deklarisana).

```
<varijabla kojoj dodeljujemo vrednost> = <izraz>;
broj = 5;
Tekst = "Zdravo";
josJedanBroj = 5.5;
drugibroj = 3 + broj;
```

Primeri naredbi dodele

Da bi varijablama mogli da dodelimo neke smislene vrednosti moramo da znamo da pišemo izraze, izrazi u programiranju su isto što i izrazi u matematici, niz vrednosti i operacija koje vršimo nad njima. Kao vrednosti u izrazu možemo koristiti konstante, vrednosti nekih varijabli ili rezultate nekih izraza ili operacija.

Tip vrednosti	Izraz	Vrednost
Celobrojna konstanta	`5`	`5`
Decimalna konstanta	`3.75`	`3.75`
Tekstualna konstanta (**Napomena**: tekstualne konstante pišemo između dvostrukih navodnika `" "` da bi ih računar razlikovao od imena varijabli)	`"Zdravo svete!"`	`Zdravo svete!`
Konstanta tipa `char` (jedan karakter) (**Napomena**: `char` konstante pišemo između jednostrukih navodnika `' '` da bi ih računar razlikovao od imena varijabli)	`'A'`	`A`
Vrednost varijable	`broj`	Koja god vrednost se nalazi u varijabli broj u tom trenutku
Vrednost izraza	`3+5`	`8`
Vrednost funkcije	`Math.Max(3,5)`	`5`

Primeri izraza

+	Sabiranje
-	Oduzimanje
*	Množenje
/	Deljenje

`%`	Ostatak pri celobrojnom deljenju
`()`	Grupisanje operacija

Primeri operatora

Unos i ispis podataka u konzolu

Nakon što smo naučili da varijablama dodeljujemo vrednosti koje izračunavamo uz pomoć operatora, bilo bi zgodno da te vrednosti možemo nekako da prikažemo korisniku i da iskoristimo vrednosti koje korisnik unese za neka izračunavanja. Za to nam je potrebno da uvedemo novi koncept a to su funkcije. Funkcije uzimaju nekoliko vrednosti (ponekad i ni jednu) urade nešto sa njima i onda nam vrate neku vrednost (ili ponekad ne).

```
<Ime grupe funkcija ili varijable>.<Ime funkcije>(<Argumenti>);
Console.ReadLine();
Console.WriteLine("Zdravo svete!");
```

Primer funkcije

Ovde primećujemo nekoliko stvari:

- Grupu funkcija `Console`, nju koristimo za unos i ispis na konzolu
- Funkciju `Console.ReadLine()` koja će učitati liniju teksta sa konzole i vratiti nam njenu sadržinu
- Funkciju `Console.WriteLine()` koja kao argument uzima string, ispisuje ga na konzolu i prelazi u novi red
- Takođe postoji i funkcija `Console.Write()` koja radi slicnu stvar osim što ne prelazi u novi red
- Ukoliko funkcija ne prima argumente i dalje je neophodno imati prazne zagrade
- Ukoliko funkcija prima više od jednog argumenta onda se argumenti odvajaju zarezom (npr. `Math.Max(5,3)`)

Napomena: pošto se nakon izvršavanja poslednje komande gasi, obično se stavlja Console.ReadLine() na kraj programa da bi program ostao uključen dok se ne pritisne **enter**. (Ovo više nije neophodno u verziji 2019 Microsoft Visual Studio-a)

Konverzije

Kada bi probali da broj koji je korisnik uneo saberemo sa nekim drugim brojem dobili bi grešku je je taj broj zapravo tekst tipa string a da bi mogli da ga koristimo za sabiranje moramo da ga konvertujemo u broj. U C# razlikujemo dve vrste konverzija: implicitne i eksplicitne. Implcitne konverzije računar radi umesto nas i o njima nema potrebe da brinemo, eksplicitne konverzije su kada mi eksplicitno kažemo računaru da pretvori neku vrednost iz jednog tipa u drugi i to je ono što je neophodno da uradimo u ovom slučaju. Za konverzije koristimo grupu funkcija Convert.

```
Convert.ToInt32("532")  //vraća vrednost 532 kao broj
```

Primer eksplicitne konverzije

`ToBoolean()`	Konvertuje u **`bool`**
`ToChar()`	Konvertuje u **`char`**
`ToInt32()`	Konvertuje u **`int`**
`ToSingle()`	Konvertuje u **`float`**
`ToString()`	Konvertuje u **`string`**

1. Primeri

1. Unose se dva broja i ispisuje se njihov zbir.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            string a;
            string b;
            int ba;
            int bb;
            int bc;

            a = Console.ReadLine();
            b = Console.ReadLine();

            ba = Convert.ToInt32(a);
            bb = Convert.ToInt32(b);

            bc = ba + bb;

            Console.WriteLine(bc);
```

```
            Console.ReadLine();
        }
    }
}
```

2. Unose se vremenski interval kao broj sata i minuta kada je poceo i broj sata i minuta kada se zavrsio. Ispisati koliko sata i minuta traje interval.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            string ss;
            string sm;
            string es;
            string em;
            int bss;
            int bsm;
            int bes;
            int bem;
            int bsu;
            int beu;
            int bi;

            ss = Console.ReadLine();
            sm = Console.ReadLine();
            es = Console.ReadLine();
            em = Console.ReadLine();

            bss = Convert.ToInt32(ss);
            bsm = Convert.ToInt32(sm);
            bes = Convert.ToInt32(es);
            bem = Convert.ToInt32(em);

            bsu = bss * 60 + bsm;
            beu = bes * 60 + bem;
```

```
            bi = beu - bsu;

            Console.WriteLine(bi);

            Console.ReadLine();
        }
    }
}
```

1. Zadatci

1. Napraviti program koji unosi dva broja i prikazuje njihov proizvod na ekranu.
2. Napraviti program koji unosi tri broja i prikazuje njihov zbir na ekranu.
3. Napišite program za pretvaranje centimetara u metre. Zadaje se merni broj u centimetrima, a treba odrediti koliko je to metara.
4. Napraviti program koji računa prosek 4 uneta broja.
5. Korisnik unosi broj minuta koliko traje film, ispisati koliko je to sati i minuta.
6. Korisnik unosi u koliko sati i minuta je krenuo na autobus, u koliko sati i minuta autobus kreće i koliko minuta mu treba da stigne do stanice. Ispisati koliko minuta će on poraniti na autobus.
7. Korisnik unosi veličinu jajeta koje želi da skuva, jaje moze biti velicine od 1 do 10, jaje veličine jedan se kuva 5 minuta a za svaku jedinicu veličine preko 1 dodajemo još 2 minuta.
8. Unosi se temperatura u celzijusima, ispisati koliko je to farenhajta.
   $$T_{(°F)} = T_{(°C)} \times 1.8 + 32$$
9. Biciklista kreće da vozi bicikl u 8 ujutru, unosi se kada je završio sa treningom (kao broj sati i broj minuta) i brzina u kilometrima na čas. Ispisati koliko kilometara je prešao
10. Unosi se pravougaonik preko kordinata dve tačke, izračunati njegovu površinu i obim.
11. Unosi se trocifreni broj, zameniti njegovu cifru jedinica i stotina.
12. Osmisliti algoritam koji zamenjuje vrednosti dve varijable.

1. Rešenja

Napomena: iz rešenja je izostavljen šablon konzolne aplikacije već je naveden samo kod unutar `Main` funkcije

1.

```
int a = Convert.ToInt32(Console.ReadLine());
int b = Convert.ToInt32(Console.ReadLine());

int c = a * b;

Console.WriteLine(c);
```

2.

```
int a = Convert.ToInt32(Console.ReadLine());
int b = Convert.ToInt32(Console.ReadLine());
int c = Convert.ToInt32(Console.ReadLine());

int d = a + b + c;

Console.WriteLine(d);
```

3.

```
int cm = Convert.ToInt32(Console.ReadLine());

float m = cm / 100.0f;

Console.WriteLine(m);
```

4.

```
int a = Convert.ToInt32(Console.ReadLine());
int b = Convert.ToInt32(Console.ReadLine());
int c = Convert.ToInt32(Console.ReadLine());
int d = Convert.ToInt32(Console.ReadLine());

float p = (a+b+c+d) / 4.0f;

Console.WriteLine(p);
```

5.

```
int UkupnoMinuta = Convert.ToInt32(Console.ReadLine());

int Sati = UkupnoMinuta / 60;
int Minuti = UkupnoMinuta % 60;

Console.WriteLine(Sati+":"+Minuti);
```

6.

```
int KrenuoS = Convert.ToInt32(Console.ReadLine());
int KrenuoMin = Convert.ToInt32(Console.ReadLine());
int Vreme = Convert.ToInt32(Console.ReadLine());
int BusS = Convert.ToInt32(Console.ReadLine());
int BusMin = Convert.ToInt32(Console.ReadLine());

int KrenuoU = KrenuoS * 60 + KrenuoMin;
int BusU = BusS * 60 + BusMin;
int Poranio = BusU - (KrenuoU + Vreme);

Console.WriteLine(Poranio);
```

7.

```
int Velicina = Convert.ToInt32(Console.ReadLine());
int Vreme = Velicina * 2 + 3;

Console.WriteLine(Vreme);
```

8.

```
int TC = Convert.ToInt32(Console.ReadLine());
float TF = TC * 1.8F + 32;

Console.WriteLine(TF);
```

9.

```
int S = Convert.ToInt32(Console.ReadLine());
int M = Convert.ToInt32(Console.ReadLine());
int B = Convert.ToInt32(Console.ReadLine());

int UkupnoVreme = (S * 60 + M) - 8 * 60;
float PP = UkupnoVreme * (B / 60.0f);

Console.WriteLine(PP);
```

10.

```
int X1 = Convert.ToInt32(Console.ReadLine());
int Y1 = Convert.ToInt32(Console.ReadLine());
int X2 = Convert.ToInt32(Console.ReadLine());
int Y2 = Convert.ToInt32(Console.ReadLine());
int A = X2 - X1;
int B = Y2 - Y1;
int O = 2 * A + 2 * B;
int P = A * B;

Console.WriteLine("Obim:"+O);
Console.WriteLine("Povrsina:" + P);
```

11.

```
int B = Convert.ToInt32(Console.ReadLine());
int J = B % 10;
int D = (B / 10) % 10;
int S = B / 100;
int NB = J * 100 + D * 10 + S;

Console.WriteLine(NB);
```

12.

```
int A = Convert.ToInt32(Console.ReadLine());
int B = Convert.ToInt32(Console.ReadLine());

int c = A;
A = B;
B = c;

Console.WriteLine(A+" "+B);
```

2. Operacije sa stringovima

Pre nego što se upoznamo sa operacijama nad stringovima moramo da se upoznamo sa još dva koncepta u programiranju: nizovi i primena funkcija nad varijablama.

Nizovi su vrsta varijabli koja može da čuva više podataka istog tipa pod jednim imenom, niz možemo zamisliti kao zgradu, adresa zgrade je ime niza a ono što se kod niza zovu indeksi bi bili brojevi stanova. Za sada nećemo pričati kako se deklarišu nizovi već samo kako čitamo i upisujemo vrednosti u njih.

```
<Ime niza>[<Indeks>] = 5;           //Pisanje u niz
int x = <Ime niza>[<Indeks>];       //Citanje iz niza
string a = "JABUKA";
char c = a[2];                     //Karater c ce imati vrednost 'B'
```

Primeri čitanja i pisanja u niz

Ono što je specifično za indekse nizova jeste da počinju od nule, to jest niz od tri elementa će imati indekse: 0,1 i 2. Ovo nam je bitno za rad sa stringovima žato što su stringovi zapravo predstavljeni kao niz karaktera.

Drugi koncept koji moramo da znamo je primena funkcija nad varijablama, do sada smo pre tačke pri pozivu funkcije navodili kojoj grupi funkcija ona pripada. Druga stvar koju možemo da navedemo je ime neke varijable i onda će ta varijabla biti jedan od argumenata te funkcije.

```
string a  = "baba";
string b = a.ToUpper(); //string b ce imati vrednost "BABA"
```

Primer izvršavanja funkcije nad varijablom

Sada ćemo navesti bitne operatore i funkcije koje koristimo sa stringovima: (*s* zameniti sa string varijablom)

- *s* **+** *s*: Spaja dva stringa
- *s***[**<*Indeks*>**]**: Vraća karakter na datom indeksu u stringu
- *s*.**Length**: Vraća dužinu datog stringa
- *s*.**Substring(**<*Početni indeks*>**)**: Vraća podstring počevši od početnog indeksa
- *s*.**Substring(**<*Početni indeks*>,<*Dužina*>**)**: Vraća podstring počevši od početnog indeksa određene dužine
- *s*.**Contains(**<*Drugi string*>**)**: Vraća `true` ukoliko string sadrži drugi string, u suprotnom `false`
- *s*.**EndsWith(**<*Drugi string*>**)**: Vraća `true` ukoliko se string završava drugim stringom, u suprotnom `false`
- *s*.**StartsWith(**<*Drugi string*>**)**: Vraća `true` ukoliko string počinje drugim stringom, u suprotnom `false`

- *s*.**IndexOf(**<*Karakter ili drugi string*>**)**: Vraća poziciju prvog pojavljivanja karaktera ili drugog stringa u stringu, ako se ne pojavljuje vraća `-1`
- *s*.**IndexOf(**<*Karakter ili drugi string*>,<*Početni indeks*>**)**: Vraća poziciju prvog pojavljivanja karaktera ili drugog stringa u stringu, počevši od početnog indeksa, ako se ne pojavljuje vraća `-1`
- *s*.**LastIndexOf(**<*Karakter ili drugi string*>**)**: Vraća poziciju poslednjeg pojavljivanja karaktera ili drugog stringa u stringu, ako se ne pojavljuje vraća `-1`
- *s*.**Insert(**<*Početni indeks*>,<*Drugi string*>**)**: Vraća string dobijen umetanjem drugog stringa počev od početnog indeksa
- *s*.**PadLeft(**<*Broj karaktera*>**)**: Dodaje razmake sa leve strane stringa dok ne postane željene dužine
- *s*.**PadRight(**<*Broj karaktera*>**)**: Dodaje razmake sa desne strane stringa dok ne postane željene dužine
- *s*.**Remove(**<*Početni indeks*>**)**: Uklanja sve karaktere iz stringa od početnog indeksa do kraja
- *s*.**Remove(**<Početni indeks>,<*Dužina*>**)**: Uklanja dužina karaktera iz stringa počev od početnog indeksa
- *s*.**Replace(**<*Prvi karakter ili tekst*>,<*Drugi karakter ili tekst*>**)**: Zamenjuje sva pojavljivanja prvog karaktera ili teksta drugim karakterom ili tekstom
- *s*.**ToLower()**: Konvertuje sva slova u mala slova
- *s*.**ToUpper()**: Konvertuje sva slova u velika slova
- *s*.**Trim()**: Uklanja sve razmake sa početka i kraja stringa
- *s*.**TrimStart()**: Uklanja sve razmake sa početka stringa
- *s*.**TrimEnd()**: Uklanja sve razmake sa kraja stringa
- *s*.**Split(**<*Karakter za podelu*>**)**: Vraća niz stringova dobijen deljenjem stringa gde god se pojavljuje karakter za podelu

```
string a="a b c";
a.Split()[0]; //"a"
a.Split()[1]; //"b"
a.Split()[2]; //"c"
```

Primer Split funkcije

2. Primeri

```
string s = Console.ReadLine();
s = s + "@gmail.com";
Console.WriteLine(s);
Console.WriteLine(s.Length);
Console.WriteLine(s[s.Length-1]);
```

```
string s = Console.ReadLine();
int i = s.IndexOf(' ');
int a = Convert.ToInt32(s.Substring(0, i));
int b = Convert.ToInt32(s.Substring(i+1));
```

```
Console.WriteLine(a+b);
```

```
string s = Console.ReadLine();
string s2 = Console.ReadLine();
Console.WriteLine(s.Contains(s2));
Console.WriteLine(s.StartsWith(s2));
Console.WriteLine(s.EndsWith(s2));
```

```
string s = Console.ReadLine();
s = s.Insert(2, "f");
s = s.Remove(3, 1);
Console.WriteLine(s);
```

```
string s = Console.ReadLine();
Console.WriteLine(s.ToUpper());
Console.WriteLine(s.ToLower());
```

```
string s = Console.ReadLine();
Console.WriteLine(s.Trim());
Console.WriteLine(s.TrimStart());
Console.WriteLine(s.TrimEnd());
```

```
string str = Console.ReadLine();
int h = Convert.ToInt32(str.Split(' ')[0]);
int m = Convert.ToInt32(str.Split(' ')[1]);
int s = Convert.ToInt32(str.Split(' ')[2]);
Console.WriteLine((h*60+m)*60+s);
```

```
Console.WriteLine("15".PadLeft(4));
Console.WriteLine("1".PadLeft(4));
Console.WriteLine("123".PadLeft(4));
```

2. Zadatci

1. Unosi se ime i prezime, ispisati spojeno prvo slovo imena i prvo slovo prezimena.
2. Proveriti da li se string zavrsava istim slovom kojim pocinje.
3. Unosi se tekst koji sadrži neki drugi tekst u zagradama, potrebno je izbrisati zagrade i tekst u njima.
   Primer: Gepard (veoma brza zivotinja) zivi u Africi.
   Gepard zivi u Africi.

4. Unosi se tekst, obrisati prvu i poslednju reč u njemu.
5. Potrebno je formirati mail za zaposlenog čije se ime i prezime unose u jednom redu, Mail je u formatu: prva tri slova imena + prezime + “@gmail.com”.
6. Unosi se vremenski interval kao broj sati minuta i sekundi razdvojenih razmakom (1:15:26), ispisati koliko je to sekundi.
7. Unosi se ime i prezime u jednom redu, ispisati prvo prezime pa ime u jednom redu tako da je prezime svim malim slovima a ime svim velikim slovima.
8. Unose se tri broja u jednom redu, izračunati njihov prosek.
9. Unosi se reč, zameniti joj treće i pretposlednje slovo.

2. Rešenja

1.

```
string Ime = Console.ReadLine();
string Prezime = Console.ReadLine();
string Res = Ime.Substring(0, 1) + Prezime.Substring(0, 1);

Console.WriteLine(Res);
```

2.

```
string IN = Console.ReadLine();
bool Res = IN.EndsWith(IN.Substring(0, 1));

Console.WriteLine(Res);
```

3.

```
string IN = Console.ReadLine();
int Oz = IN.IndexOf('(');
int Dz = IN.IndexOf(')') - Oz + 2;
string Res = IN.Remove(Oz, Dz);

Console.WriteLine(Res);
```

4.

```
string IN = Console.ReadLine();
int Prvr = IN.IndexOf(' ');
int Posr = IN.LastIndexOf(' ');
string Res = IN.Substring(Prvr+1,Posr-Prvr);

Console.WriteLine(Res);
```

5.

```
string IN = Console.ReadLine();
string ime = IN.Substring(0, IN.IndexOf(' '));
string prezime = IN.Substring(IN.IndexOf(' ') + 1);
string Res = ime.Substring(0, 3) + prezime + "@gmail.com";

Console.WriteLine(Res);
```

6.

```
string IN = Console.ReadLine();
int h = Convert.ToInt32(IN.Split(':')[0]);
int m = Convert.ToInt32(IN.Split(':')[1]);
int s = Convert.ToInt32(IN.Split(':')[2]);
int us = (h * 60 + m) * 60 + s;
Console.WriteLine(us);
```

7.

```
string IN = Console.ReadLine();
string ime = IN.Substring(0, IN.IndexOf(' '));
string prezime = IN.Substring(IN.IndexOf(' ')+1);
string Res = prezime.ToLower() + " " + ime.ToUpper();

Console.WriteLine(Res);
```

8.

```
string IN = Console.ReadLine();
int a = Convert.ToInt32(IN.Split(' ')[0]);
int b = Convert.ToInt32(IN.Split(' ')[1]);
int c = Convert.ToInt32(IN.Split(' ')[2]);
float avg = (a + b + c) / 3.0f;

Console.WriteLine(avg);
```

9.

```
string IN = Console.ReadLine();

string p = IN.Substring(0, 2);
char t = IN[2];
string m = IN.Substring(3, IN.Length - 5);
char pp = IN[IN.Length - 2];
string k = IN.Substring(IN.Length - 1);
string Res = p + pp + m + t + k;

Console.WriteLine(Res);
```

3. Naredbe grananja

Do sada su se programi koje smo pisali izvršavali inijski odozgo na dole i svaka komanda je bila izvršena, uvođenje naredbi grananja nam odmogućava da imamo delove koda koji se izvršavaju samo ako je neki logički uslov ispunjen.

Logički izrazi

Da bi mogli da formiramo uslov koji određuje da li se deo koda izvršava potrebno je da zamo da pišemo logičke izraze. Logički izrazi nisu mnogo drugačiji od izraza sa kojima smo se do sada sretali, jedina razlika je to da logički izraz na kraju vraća logičku vrednost (`true` ili `false`) i da postoje posebni operatori za rad sa logičkim vrednostima.

Logički operatori:

==	Jednakost, radi na dva operanda bilo kog tipa
!=	Nije jednako, radi na dva operanda bilo kog tipa
<	Manje, radi na dva operanda brojevnog tipa
<=	Manje jednako, radi na dva operanda brojevnog tipa
>	Veće, radi na dva operanda brojevnog tipa
>=	Veće jednako, radi na dva operanda brojevnog tipa
!	Nije, radi na jednom operandu logičkog tipa (bool)
&&	I, radi na dva operanda logičkog tipa
||	Ili, radi na dva operanda logičkog tipa

A	B	!A	A && B	A || B
true	true	false	true	true
true	false	false	false	true
false	true	true	false	true
false	false	true	false	false

Tablica logičkih operacija

Blok koda

Blok koda nam omogućava da više komandi predstavimo kao jednu za potrebe nekih naredbi (npr. if)

```
{
    //naredbe
}
```

Napomena: tačka zarez se ne piše nakon naredbi posle kojih dolazi blok koda.

Naredba grananja if

Naredba grananja if je naj osnovnija naredba grananja i ona u svom osnovnom obliku izvršava naredbe unutar nje samo ako je njen logički uslov tačan.

```
if(<Logički uslov>)
{
      //naredbe
}
```

Primer if naredbe

Klauza else

Else klauza nam omogućava da dopunimo našu if naredbu naredbama koje trebaju da se izvrše ako logički uslov nije tačan.

```
if(<Logički uslov>)
{
      //naredbe A
}
else
{
      //naredbe B
}

/*
if(<Logički uslov>)
{
      //naredbe A
}
if(!<Logički uslov>)
{
      //naredbe B
}
*/
```

Primer else klauze i kod koji ona zamenjuje

Klauza else if

Else if klauza nam omogućava da else klauzu proširimo dodatnim uslovom, samo ako nijedan od else if uslova nije tačan else klauza će se izvršiti.

```
if(<Logički uslov 1>)
{
	//naredbe A
}
else if(<Logički uslov 2>)
{
	//naredbe B
}
else
{
	//naredbe C
}

/*
if(<Logički uslov 1>)
{
	//naredbe A
}
else
{
	if(<Logički uslov 2>)
	{
		//naredbe B
	}
	else
	{
		//naredbe C
	}
}
*/
```

Primer else if klauze i kod koji ona zamenjuje

Naredba grananja switch

Kada nam je potrebno da promenimo tok koda u zavisnosti od neke konkretne vrednosti izraza i kada ima mnogo tih vrednosti zgodnije je koristiti switch naredbu nego nizati mnogo if naredbi.

```
switch(<Izraz>)
{
case <Vrednost1> : { /*naredbe A*/ break; }
case <Vrednost2> : { /*naredbe B*/ break; }
case <Vrednost3> : { /*naredbe C*/ break; }
default : { /*naredbe D*/ break; }
}

/*
if(<Izraz> == <Vrednost1>)
{
      //naredbe A
}
else if(<Izraz> == <Vrednost2>)
{
      //naredbe B
}
else if(<Izraz> == <Vrednost3>)
{
      //naredbe C
}
else
{
      //naredbe D
}
*/
```

Primer switch naredbe i kod koji ona zamenjuje

Naredba try/catch

try/catch naredba nam omogućava da obradimo greške koje se dešavaju prilikom pokretanja programa (na primer: ako pokušamo da pretvorimo “Tekst” u broj desiće se greška prilikom izvršavanja). Try catch će izvršiti komande try bloka i ako se u try bloku desi greška umesto njih će izvršiti catch blok.

```
try
{
      //komande za try blok
}
catch
{
      //komande koje se izvrše ako se desi greška u try blok-u
}
```

Primer try/catch naredbe

3. Primeri

1. Maksimum dva broja.

```
int a = Convert.ToInt32(Console.ReadLine());
int b = Convert.ToInt32(Console.ReadLine());
if (a < b)
{
      Console.WriteLine(b);
}
else
{
      Console.WriteLine(a);
}
```

2. Absolutna vrednost.

```
int a = Convert.ToInt32(Console.ReadLine());

if (a < 0)
{
      a = -a;
}

Console.WriteLine(a);
```

3. Kalkulator sa dve operacije.

```
int a = Convert.ToInt32(Console.ReadLine());
int b = Convert.ToInt32(Console.ReadLine());
string o = Console.ReadLine();

if (o == "+")
{
      Console.WriteLine(a+b);
}
else if (o == "-")
{
      Console.WriteLine(a-b);
}
else
{
      Console.WriteLine("Nije zadata tacna operacija");
}
```

4. Program koji ispisuje puno ime strane sveta na osnovu prvog slova.

```
string input = Console.ReadLine();
switch (input.ToUpper())
```

```
{
case "I": { Console.WriteLine("Istok"); break; }
case "Z": { Console.WriteLine("Zapad"); break; }
case "S": { Console.WriteLine("Sever"); break; }
case "J": { Console.WriteLine("Jug"); break; }
default: { Console.WriteLine("Ne postoji to pocetno slovo"); break; }
}
```

3. Zadatci

1. Biciklista Zdravko planira da kupi novi bicikl kako bi celo leto vozio krugove po Adi, dobio je ponudu od Mileta da povoljno kupi bicikl, prijatelj mu nudi dva bicikla po različitim cenama a Zdravko planira da kupi jeftiniji pošto su oba odgovarajućeg kvaliteta. Napišite program koji će Zdravku da pomogne u izboru.
2. Zdravko je kupio novi bicikl ali i dalje nije u dobroj formi za vožnju, Mile mu je predložio da idu da se provozaju za vikend i ponudio je tri različite putanje. Pošto Zdravko nije u formi napišite program koji će mu pomoći da odabere turu sa najmanje kilometara.
3. Posle par treninga Zdravko je počeo redovno da vozi ali zbog obaveza ne može svaki dan da trenira. Ako mu dnevna vožnja iznosi 20km i nikad više od toga, napišite program koji računa koliko dana u nedelji Zdravko mora da vozi bicikl kako bi prešao bar X kilometara, gde se X unosi. Ako je nemoguće da se za nedelju dana pređe X kilometara ispisati odgovarajuću poruku.
4. Tokom jedne od vožnji sa Raletom naišli su na raskrsnicu i tri moguća puta do kuće (A,B i C). Pošto obojica hoće što više da treniraju taj dan, napisati program koji će odabrati za njih najduži put od tri uneta.
5. Zdravko prilikom planiranja vožnje za sledeći vikend ima dilemu, ako je vožnja kraća od 50km hteo bi da vozi po brdu a ako je duža voziće po ravnici. Zatim kada odabere put treba da odluči koliko vode da ponese pošto nigde usput nema nijedna česma. Ako ide po brdu trebaće mu 1 litar za svakih 5km, a ako ide po ravnom treba mu 1 litar za svakih 10 km. Takođe ako ide u brdsku vožnju poneće fotoaparat da slika veverice u šumi a ako ide po ravnom treba da ponese mrvice da nahrani patke na reci. Napišite program koji za unetu kilometražu ispisuje putanju, količinu vode i stvari koje Zdravko nosi sa sobom.
6. Napisati program koji proverava da li je uneta godina prestupna (deljiva sa 4 i ne sa 100 ili deljiva sa 400).
7. Napisati program koji za uneti karaker proverava da li je veliko ili malo slovo.
8. Napisati program koji za uneta tri ugla trougla proverava da li taj trougao postoji.
9. Napisati program koji za unetu tačku (unose se X i Y koordinate) pronalazi u kom kvadrantu se nalazi ta tačka.
10. Unosi se zaključna ocena (int), ispisati tekstualni opis ocene (odlican,vrlo dobar,...).
11. Kalkulator sa sve 4 operacije (opciono ubaciti try catch).

3. Rešenja

1.

```
int a = Convert.ToInt32(Console.ReadLine());
int b = Convert.ToInt32(Console.ReadLine());
if (a < b)
{
    Console.WriteLine(a);
}
else
{
    Console.WriteLine(b);
}
```

2.

```
int a = Convert.ToInt32(Console.ReadLine());
int b = Convert.ToInt32(Console.ReadLine());
int c = Convert.ToInt32(Console.ReadLine());
if (a < b)
{
    if (a < c)
    {
        Console.WriteLine(a);
    }
    else
    {
        Console.WriteLine(c);
    }
}
else
{
    if (b < c)
    {
        Console.WriteLine(b);
    }
    else
    {
        Console.WriteLine(c);
    }
}
```

3.

```
int a = Convert.ToInt32(Console.ReadLine());
if (a < 140)
{
```

```
    Console.WriteLine((a + 19) / 20);
}
else
{
    Console.WriteLine("Nije moguce da postigne zadatu kilometrazu");
}
```

4.

```
int a = Convert.ToInt32(Console.ReadLine());
int b = Convert.ToInt32(Console.ReadLine());
int c = Convert.ToInt32(Console.ReadLine());
if (a > b && a > c)
{
    Console.WriteLine(a);
}
else if (b > a && b > c)
{
    Console.WriteLine(b);
}
else
{
    Console.WriteLine(c);
}
```

5.

```
int a = Convert.ToInt32(Console.ReadLine());
if (a < 50)
{
    Console.WriteLine("Planina");
    Console.WriteLine((a + 4) / 5);
    Console.WriteLine("Fotoaparat");
}
else
{
    Console.WriteLine("Ravnica");
    Console.WriteLine((a + 9) / 10);
    Console.WriteLine("Mrvice");
}
```

6.

```
int a = Convert.ToInt32(Console.ReadLine());
if ((a % 4 == 0 && a % 100 != 0) || a % 400 == 0)
{
    Console.WriteLine("Prestupna");
}
else
```

```
{
    Console.WriteLine("Nije prestupna");
}
```

7.

```
string s = Console.ReadLine();
if (s.ToUpper() == s)
{
    Console.WriteLine("Veliko");
}
else
{
    Console.WriteLine("Malo");
}
```

8.

```
int a = Convert.ToInt32(Console.ReadLine());
int b = Convert.ToInt32(Console.ReadLine());
int c = Convert.ToInt32(Console.ReadLine());
if (a + b + c == 180)
{
    Console.WriteLine("Postoji");
}
else
{
    Console.WriteLine("Ne postoji");
}
```

9.

```
int x = Convert.ToInt32(Console.ReadLine());
int y = Convert.ToInt32(Console.ReadLine());
if (x >= 0)
{
    if (y >= 0)
    {
        Console.WriteLine(1);
    }
    else
    {
        Console.WriteLine(4);
    }
}
else
{
    if (y >= 0)
    {
```

```
        Console.WriteLine(2);
    }
    else
    {
        Console.WriteLine(3);
    }
}
```

10.

```
int a = Convert.ToInt32(Console.ReadLine());
switch (a)
{
    case 1: { Console.WriteLine("Nedovojan"); break; }
    case 2: { Console.WriteLine("Dovjoan"); break; }
    case 3: { Console.WriteLine("Dobar"); break; }
    case 4: { Console.WriteLine("Vrlo dobar"); break; }
    case 5: { Console.WriteLine("Odlican"); break; }
    default: { Console.WriteLine("Ocena mora biti <= 5"); break; }
}
```

11.

```
try
{
    int a = Convert.ToInt32(Console.ReadLine());
    int b = Convert.ToInt32(Console.ReadLine());
    string op = Console.ReadLine();
    switch (op)
    {
        case "+": { Console.WriteLine(a+b); break; }
        case "-": { Console.WriteLine(a-b); break; }
        case "/": { Console.WriteLine(a/b); break; }
        case "*": { Console.WriteLine(a*b); break; }
        default: { Console.WriteLine("Ta operacija ne postoji"); break;
}
    }
}
catch
{
    Console.WriteLine("Pogresan unos");
}
```

4. Naredbe Ponavljanja

Do sada su se sve naredbe koje smo pisali izvršavale tačno jednom ili nijednom (korišćenjem naredbi grananja), ali nekada je potrebno da se neke naredbe izvrše više puta ili da se izvršavaju dokle god je neki uslov ispunjen.

Naredba ponavljanja while

While je osnovna naredba ponavljanja, sastoji se od uslova i bloka koda. While naredba izvršava blok koda dokle god je uslov ispunjen.

```
while(uslov)
{
    //kod
}
```

Primer while naredbe

Naredba ponavljanja for

For je naredba slična while naredbi ali se uglavnom primenjuje kada neki kod želimo da izršimo tačan broj puta.

```
for(inicijalizacija; uslov; korekcija)
{
      //kod
}

/*
Inicijalizacija;
while(uslov)
{
      //kod

korekcija;
}
*/
```

Primer for naredbe i ekvivalentne while naredbe

Trikovi za skraćivanje koda

int a=3; int b=5;	int a=3,b=5;

a = a + 3;	a +=3;
a = a - 3;	a -=3;
a +=1;	a++;
a -=1;	a--;

4. Primeri

1. Ispisati prvih 10 prirodnih brojeva.
   a. While

```
int i = 0;
while (i < 10)
{
    Console.WriteLine(i+1);
    i++;
}
```

   b. For

```
for (int i = 0; i < 10; i++)
{
    Console.WriteLine(i+1);
}
```

2. Ispisati sumu prvih 10 prirodnih brojeva.
   a. While

```
int i = 0,sum=0;
while (i < 10)
{
    sum += i + 1;
    i++;
}
Console.WriteLine(sum);
```

   b. For

```
int sum = 0;
for (int i = 0; i < 10; i++)
{
    sum += i + 1;
}
Console.WriteLine(sum);
```

4. Zadatci

1. Ispisati program koji ispisuje prvih n prirodnih brojeva.
2. Ispisati program koji ispisuje sumu prvih n prirodnih brojeva.

3. Ispisati program koji ispisuje sumu i prosek n unetih brojeva.
4. Ispisati program koji ispisuje kubove prirodnih brojeva do zadatog broja.
   Primer: 125 1 8 27 64 125
5. Ispisati program koji ispisuje tablicu množenja do 10 unetog broja.
   Primer: 7 7 14 21 28 35 42 49 56 63 70
6. Ispisati program koji ispisuje prvih n neparnih brojeva i njihovu sumu.
7. Ispisati program koji ispisuje tablicu množenja do n.
   Primer: 3 1 2 3
   2 4 6
   3 6 9
8. Ispisati program koji iscrtava dati šablon.
   a. *
      **
      ***
      ****
   b. 1
      12
      123
      1234
   c. 1
      22
      333
      4444
   d. 1
      2 3
      4 5 6
      7 8 9 10
   e. *
       * *
      * * *
     * * * *
9. Ispisati program koji ispisuje fakrorijal unetog broja.
10. Ispisati program koji iscrtava šahovsku tablu.

4. Rešenja

1.

```
int n = Convert.ToInt32(Console.ReadLine());
for (int i = 0; i < n; i++)
{
    Console.WriteLine(i + 1);
}
```

2.

```
int n = Convert.ToInt32(Console.ReadLine());
int sum = 0;
for (int i = 0; i < n; i++)
{
    sum += i + 1;
}
Console.WriteLine(sum)
```

3.

```
int n = Convert.ToInt32(Console.ReadLine());
int sum = 0;
for (int i = 0; i < n; i++)
{
    sum += i + 1;
}
Console.WriteLine(sum);
Console.WriteLine(sum / (float)n);
```

4.

```
int n = Convert.ToInt32(Console.ReadLine());
for (int i = 0; i * i * i <= n; i++)
{
    Console.WriteLine(i * i * i);
}
```

5.

```
 int n = Convert.ToInt32(Console.ReadLine());
for (int i = 1; i <= 10; i++)
{
    Console.WriteLine(n * i);
}
```

6.

```
int n = Convert.ToInt32(Console.ReadLine());
for (int i = 0; i < n; i++)
{
```

```
    Console.WriteLine((2 * i) + 1);
}
```

7.

```
int n = Convert.ToInt32(Console.ReadLine());
for (int i = 1; i <= n; i++)
{
    for (int j = 1; j <= n; j++)
    {
        Console.Write(i * j + " ");
    }
    Console.WriteLine();
}
```

8.

a.

```
for (int i = 0; i < 4; i++)
{
    for (int j = 0; j <= i; j++)
    {
        Console.Write("*");
    }
    Console.WriteLine();
}
```

b.

```
for (int i = 0; i < 4; i++)
{
    for (int j = 0; j <= i; j++)
    {
        Console.Write(j + 1);
    }
    Console.WriteLine();
}
```

c.

```
for (int i = 0; i < 4; i++)
{
    for (int j = 0; j <= i; j++)
    {
        Console.Write(i + 1);
    }
    Console.WriteLine();
}
```

d.

```
int p = 1;
```

```
for (int i = 0; i < 4; i++)
{
    for (int j = 0; j <= i; j++)
    {
        Console.Write(p + " ");
        p++;
    }
    Console.WriteLine();
}
```

e.

```
for (int i = 0; i < 4; i++)
{
    for (int j = 0; j < 3 - i; j++)
    {
        Console.Write(" ");
    }
    for (int j = 0; j <= i; j++)
    {
        Console.Write("* ");
    }
    Console.WriteLine();
}
```

9.

```
int n = Convert.ToInt32(Console.ReadLine());
int fact = 1;
for (int i = 2; i <= n; i++)
{
    fact *= i;
}
Console.WriteLine(fact);
```

10.

```
string A, B;
for (int i = 0; i < 8; i++)
{
    if (i % 2 == 0)
    {
        A = "O";
        B = "X";
    }
    else
    {
        A = "X";
        B = "O";
```

```
    }
    for (int j = 0; j < 8; j++)
    {
        if (j % 2 == 0)
        {
            Console.Write(A);
        }
        else
        {
            Console.Write(B);
        }
    }
    Console.WriteLine();
}
```

5. Grafički interfejs

Prvi korak pri izradi aplikacija sa grafičkim interfejsom je dizajn samog interfejsa. Čim kreiramo aplikaciju sa grafičkim interfejsom u programerskom okruženju Microsoft Visual Studio biće nam prikazana prazna forma na koju mozemo da postavljamo kontrole iz kutije sa alatima.
Nakon toga u panelu sa svojstvima možemo da podesimo dodatna svojstva kontrolama koje smo postavili na formu.
Neke od kontrola sa kojima ćemo se susretati na sledećim časovima:

- **TextBox** (polje za unos teksta)
- **Button** (dugme)
- **Label** (prikaz teksta bez unosa)
- **Timer** (izvršava funkciju u pravilnim vremenskim intervalima)
- **CheckBox** (polje za unos logičke vrednosti)
- **ListBox** (prikaz liste tekstova)

Svojstva koja smo podesavali u panelu sa svojstvima takođe možemo (i često će biti neophodno) da podesimo iz koda.
Neka od svojstava i funkcija koje ćemo koristiti na sledećim časovima:

- Text (Tekst kontrole)
- Width (Širina kontrole)
- Height (Visina kontrole)
- BackColor (Boja pozadine kontrole)
- Enabled (Menja stil tako da korisnik ne može da interaguje sa kontrolom)
- ReadOnly (Zabranjuje korisniku unos u kontrolu)
- Checked (Za CheckBox, da li je štikliran)
- Items.Add() (Dodaje element u ListBox)
- Items.Clear() (Briše sve elemente iz ListBox-a)
- SelectedIndex (Pozicija selektovanog elementa u ListBox-u)
- SelectedItem (Tekst selektovanog elementa u ListBox-u)
- Start() (Započinje tajmer)
- Stop() (Zaustavlja tajmer)

Koncept i kreiranje događaja

Za razliku od konzolnih aplikacija koje imaju glavnu funkciju koja se izvršava, aplikacije sa grafičkim interfejsom imaju više funkcija koje se izvršavaju u određenim uslovima tj. događaje.
Događaji se kreiraju u panelu sa svojstvima ili duplim klikom na kontrolu.
Neki od događaja koje ćemo koristiti:

- Form_Load (Dešava se pri pokretanju programa)
- Click (Dešava se kada se klikne na neku kontrolu)
- MouseClick (Dešava se kada se klikne mišem, sadrži poziciju miša kao argument)
- Timer_Tick (Dešava se u pravilnim vremenskim intervalima)

- CheckBox_CheckedChanged (Dešava se kada se promeni štikliranost CheckBox-a)
- ListBox_SelectedIndexChanged (Dešava se kada se promeni selektovani element u ListBox-u)

Korisne funkcije

- Random (Nasumični brojevi)

```
Random r = new Random();
r.Next(min,max);
```

- Color (Boje)

```
Color.FromARGB(R,G,B);
Color.Red;
Color.Blue;
```

- MessageBox (Prikazuje dijalog sa tekstom)

```
MesageBox.Show("Tekst");
```

- DateTime (Operacije sa datumom i vremenom)

```
Datetime dt = DateTime.Now;
dt.ToShortTimeString();
dt.ToLongTimeString();
dt.ToShortDateString();
dt.ToLongDateString();
```

5. Primeri

5.Zadatci

1. Kreirajte aplikaciju kojom se vrši razmena sadržaja dve TextBox kontrole. Razmena se dešava klikom na dugme.
2. Kreirati aplikaciju kojom se klikom na dugme određuje celobrojni količnik i ostatak pri deljenju dva cela broja. Unos celih brojeva realizovati preko TextBox-a a prikaz preko jednog ili dva Label-a.
    a. Dodati proveru da li su uneti podaci ispravni .
    Hint: Try Catch i MessageBox.
3. Kreirati aplikaciju koja obezbeđuje izvršavanje osnovnih artimetičkih operacija (+,-,*,/) nad realnim brojevima. Unos realinh brojeva i prikazivanje rezultata realizovati korišćenjem TextBox-ova (TextBox za rezultat bi trebao da ima Enabled false), a izbor operacija korišćenjem 4 dugmeta. Poželjno je ali nije neophodno kontrolisati ispravnost podataka.

    Timer, DateTime...
4. Kreirati aplikaciju kojom se prikazuje sistemski datum i vreme koje se ažururia svake sekunde.

5. Kreirati aplikaciju sa jednim TextBox-om u kome je prikazano trenutno stanje brojača i 4 dugmeta:
   Napred: povećava brojač za 1 u pravilnim vremenskim intervalima.
   Nazad: smanjuje brojač za 1 u pravlinim vremenskim intervalima.
   Zaustavi: zaustavlja brojanje.
   Poništi: zaustavlja brojanje i vraća brojač na 0.

   ListBox i CheckBox
6. Kreirati aplikaciju koja ime i prezime, uneto u polje za unos teksta. Klikom na dugme ime i prezime se razdvajaju i dodaju se u dve ListBox kontrole
7. Kreirati aplikaciju koja sadrži ListBox kontrolu u kojoj se nalaze geometrijski oblici. Klikom na neki od oblika, ukoliko je CheckBox štikliran treba u Label-u ispisati formulu za površinu datog tela a ukoliko nije štikliran treba ispisati formulu za obim.
8. Kreirati aplikaciju koja u ListBox upisuje sve parne ili neparne brojeve (zavisi od Check Box-a) do broja unetog u TextBox.
9. Kreirati aplikaciju koja za dati broj u TextBox-u klikom na dugme u ListBox dodaje sve njegove delioce.

5. Rešenja

1.

```
private void button1_Click(object sender, EventArgs e)
{
    string s = textBox1.Text;
    textBox1.Text = textBox2.Text;
    textBox2.Text = s;
}
```

2.

```
private void button1_Click(object sender, EventArgs e)
{
    try
    {
        int a = Convert.ToInt32(textBox1.Text);
        int b = Convert.ToInt32(textBox2.Text);
        label1.Text = "Kolicnik je:" + (a / b) + " Ostatak :" + (a % b);
    }
    catch
    {
        MessageBox.Show("Pogresan Unos");
    }
}
```

3.

```
private void button1_Click(object sender, EventArgs e)
{
    int a = Convert.ToInt32(textBox1.Text);
    int b = Convert.ToInt32(textBox2.Text);
    textBox3.Text = (a + b).ToString();
}

private void button2_Click(object sender, EventArgs e)
{
    int a = Convert.ToInt32(textBox1.Text);
    int b = Convert.ToInt32(textBox2.Text);
    textBox3.Text = (a - b).ToString();
}

private void button3_Click(object sender, EventArgs e)
{
    int a = Convert.ToInt32(textBox1.Text);
    int b = Convert.ToInt32(textBox2.Text);
    textBox3.Text = (a * b).ToString();
```

```
}

private void button4_Click(object sender, EventArgs e)
{
    int a = Convert.ToInt32(textBox1.Text);
    int b = Convert.ToInt32(textBox2.Text);
    textBox3.Text = (a / b).ToString();
}
```

4.

```
private void Form1_Load(object sender, EventArgs e)
{
    label1.Text = DateTime.Now.ToLongTimeString();
    timer1.Start();
}

private void timer1_Tick(object sender, EventArgs e)
{
    label1.Text = DateTime.Now.ToLongTimeString();
}
```

5.

```
int brojac = 0;
int smer = 1;
private void timer1_Tick(object sender, EventArgs e)
{
    brojac += smer;
    textBox1.Text = brojac.ToString();
}

private void btnNapred_Click(object sender, EventArgs e)
{
    smer = 1;
    timer1.Start();
}

private void btnNazad_Click(object sender, EventArgs e)
{
    smer = -1;
    timer1.Start();
}

private void btnZaustavi_Click(object sender, EventArgs e)
{
    timer1.Stop()
}
```

```
private void btnPonisti_Click(object sender, EventArgs e)
{
    timer1.Stop();
    brojac = 0;
    textBox1.Text = brojac.ToString();
}
```

6.

```
private void button1_Click(object sender, EventArgs e)
{
    int ind = textBox1.Text.IndexOf(' ');
    listBox1.Items.Add(textBox1.Text.Substring(0, ind));
    listBox2.Items.Add(textBox1.Text.Substring(ind + 1));
}
```

7.

```
private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    switch (listBox1.SelectedIndex)
    {
        case 0: { label1.Text = checkBox1.Checked ? "r*r*pi" : "2*r*pi";
break; }
        case 1: { label1.Text = checkBox1.Checked ? "a*ha" : "a+b+c";
break; }
        case 2: { label1.Text = checkBox1.Checked ? "a*b" : "2*(a+b)";
break; }
    }
}

private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
    listBox1_SelectedIndexChanged(sender, e);
}

private void Form1_Load(object sender, EventArgs e)
{
    listBox1.Items.Add("Krug");
    listBox1.Items.Add("Trougao");
    listBox1.Items.Add("Pravougaonik");
}
```

8.

```
private void button1_Click(object sender, EventArgs e)
{
    listBox1.Items.Clear();
```

```
    int n = Convert.ToInt32(textBox1.Text);
    int p = checkBox1.Checked ? 1 : 0;
    for (int i = p; i <=n; i+=2)
    {
        listBox1.Items.Add(i);
    }
}
```

9.

```
private void button1_Click(object sender, EventArgs e)
{
    int n = Convert.ToInt32(textBox1.Text);
    for (int i = 1; i <= n; i++)
    {
        if (n % i == 0)
        {
            listBox1.Items.Add(i);
        }
    }
}
```

6. Klasa Graphics

Klasa Graphics nam omogućava da direktno manupulišemo pikselima forme i crtamo oblike po njoj.

Kordinatni sistem koji koristimo za iscrtavanje ima horizontalnu x osu i vertikalnu y osu, centar koordinatnog sistema je gornji levi ugao (x osa raste na desno a y na dole), donji desni ugao forme ima koordinate koje zavise od veličine forme a možemo ih pronaći u varijablama **ClientRectangle.Width** i **ClientRectangle.Height**.

Objekat kase Graphics kreiramo sledećom naredbom: `Graphics g = CreateGraphics();`
Takođe je neophodno da kada završimo rad sa Graphics objektom oslobodimo njegove resurse pozivom naredbe: `g.Dispose();`

Rad sa Color klasom je objašnjen ranije i nećemo se vraćati na njega osim da se podsetimo da postoje predefinisane boje i funkcija FromARGB() kojom specifikujemo bilo koju drugu boju.

Olovke i četke su neophodni argumenti funkcija za iscrtavanje jer oni sadrže boju (i u slučaju olovke i debljinu linije) kojom funkcija iscrtava svoj oblik, olovke se koriste kao argumenti funckija koje iscrtavaju samo konturu oblika (pa zato i sadrže informaciju o debljini linije) a četke (preciznije klasa SolidBrush) se koriste kao argumenti funkcija koje iscrtavaju popunjene oblike.
Kreiraju se na sledeći način:

```
Pen p = new Pen(<boja olovke>,<debljina linije>);
SolidBrush sb = new SolidBrush(<boja cetke>);
```

Pri kreiranju olovke drugi argument je opcioni, ako se izostavi podrazumevana debljina linije je jedan piksel.

Funkcije koje koristimo za iscrtavanje oblika na formi:

- g.Clear(<boja>)
- g.DrawLine(<olovka>,<pocetno x>,<pocetno y>,<kranje x>,<kranje y>)
- g.DrawRectangle(<olovka>,<levi x>,<gornji y>,<širina>,<visina>)
- g.DrawEllipse(<olovka>,<levi x>,<gornji y>,<širina>,<visina>)
- g.FillRectangle(<solid brush>,<levi x>,<gornji y>,<širina>,<visina>)
- g.FillEllipse(<solid brush>,<levi x>,<gornji y>,<širina>,<visina>)

6. Primeri

6. Zadatci

1. Kreirati aplikaciju koja kada korisnik klikne mišem iscrtava krug centriran na pokazivaču miša poluprečnika koji se unosi u TextBox.
    a. Dodati da se na desni klik miša briše forma.
2. Kreirati aplikaciju koja omogućava crtanje linija povlačenjem miša.
    a. Unaprediti aplikaciju da se crtaju pravougaonici.
3. Koncerntricni krugovi

6. Rešenja