

C# PROGRAMSKI JEZIK

Osnove sintakse

- Naredba predstavlja iskaz kojim se zadaje komanda računaru da izvrši neku operaciju. Kraj svake naredbe završava se simbolom tačka-zarez (";").
- Naredbe se mogu pisati jedna za drugom u istom redu, ali najčešće se pišu u posebnim redovima.
- Naredbe se pišu u blokovima oivičenim vitičastim zagradama { }. Blokovi određuju oblast vidljivosti promenljivih, odnosno oblast dejstva nekog koda. Moguće je da postoji više ugnježenih blokova.
- Programski kod C# jezika razlikuje velika i mala slova ("case sensitive" – ime i Ime nisu iste promenljive).
- Komentari se koriste da bi se neki deo programskog koda objasnio. Koristimo // za svaki red ili /* */ za blok.

Tipovi podataka

- **int** – brojučana vrednost
- **float** – decimalna brojučana vrednost
- **double** – decimalna brojučana vrednost (precizniji od float)
- **char** – jedan karakter
- **string** – niz karaktera, odnosno tekst
- **bool** – logički tip, odnosno provera tačnosti

Promenljive

Promenljive predstavljaju nazive koji su dati memorijskim lokacijama u kojima se smeštaju podaci. Podaci u tim promenljivima se u toku izvršavanja programa mogu menjati. Promenljive su određene imenom, memorijskom lokacijom, tipom podatka i vrednošću.

Pravila prilikom deklaracije/definicije promenljive :

- Naziv promenljive mora početi ili slovom ili donjom crtom, a nakon toga u sebi može sadržati slova, brojeve i donje crte.
- Nije dozvoljen razmak unutar naziva promenljive.
- Rezervisane reči poput int, float, using se ne mogu koristiti kao nazivi promenljivih.
- Kada je promenljiva sa određenim nazivom deklarirana taj naziv se više ne može koristiti za deklarisanje novih promenljivih.

Razlikujemo:

- Deklaracija promenljive – određivanje imena i tipa podatka promenljive.

```
int broj;  
double decimalanBroj;  
char karakter;  
string tekst;  
bool proveraTacnosti;
```

- Definicija promenljive – uključuje deklaraciju i dodeljivanje početne vrednosti.

```
int broj = 10;  
double decimalanBroj = 0.7;  
char karakter = 'A'; (piše se između po jednog navodnika)  
string tekst = "Pise se izmedju navodnika";  
bool proveraTacnosti = true;
```

- Deklaracija više promenljivih istog tipa podataka

```
int x, y, z;
```

- Definicija više promenljivih istog tipa podataka

```
int x=1, y=2, z=3;
```

Naredbe

jedna naredba - naredba;
blok naredbi - { blok naredbi }

Operatori

+ sabiranje, spajanje stringova, spajanje stringova i promenljivih
- oduzimanje
***** množenje
/ deljenje
% ostatak od deljenja
+= $x+=2$ je isto što i $x=x+2$
-= $x-=2$ je isto što i $x=x-2$
***=** $x*=2$ je isto što i $x=x*2$
/= $x/=2$ je isto što i $x=x/2$
%= $x%=2$ je isto što i $x=x%2$
++ $x++$ je isto što i $x=x+1$
-- $x--$ je isto što i $x=x-1$
== jednakost - vraća vrednost *tačno* ako su vrednosti promenljivih identične
!= nije jednako - vraća vrednost *tačno* ako su vrednosti promenljivih različite
< manje - vrednost izraza je *tačna* ako je promenljiva sa leve strane znaka manja od one sa desne strane
> veće - vrednost izraza je *tačna* ako je promenljiva sa leve strane znaka veća od one sa desne strane
<= manje ili jednako - vrednost izraza je *tačna* ako je promenljiva sa leve strane znaka manja ili jednaka od one sa desne strane
>= veće ili jednako - vrednost izraza je *tačna* ako je promenljiva sa leve strane znaka veća ili jednaka od one sa desne strane
&& uslovno logičko AND - ako je prvi operand *netačno*, drugi se ni ne proverava
|| uslovno logičko OR - ako je prvi operand *tačno*, drugi se ni ne proverava
t ? x : y ako je t tačno, prover i vrati vrednost x, inače prover i vrati vrednost y

Ispisivanje na ekran

Console.Write("Tekst"); ispisuje se sve između navodnika.

Console.WriteLine("Tekst"); ispisuje se sve između navodnika dok se naredni ispis prebacuje u sledeći red.

Console.WriteLine("{0} je veće od {1} dok je {2} jednako {3}", a, b, c, d);

Ispisuje vrednost promenljivih a, b, c i d redosledom kojim su nabrojane nakon zatvorenih navodnika i zareza, a na mestima brojeva ovičenih vitičastom zagradom unutar navodnika.

Console.WriteLine("Rezultat je: " + zbir);

Ispisuje vrednost promenljive zbir nakon teksta između navodnika.

Učitavanje iz konzole

Console.ReadLine();

Učitava sve što korisnik ispiše u konzolu pre nego što pritisne *Enter* (u kom god da je obliku čita ga kao promenljivu tipa string).

Console.ReadKey();

Čeka da korisnik pritisne neko dugme pre nego što nastavi sa izvršavanjem naredbi ili zatvaranjem programa.

Transformacija tipova podataka

int.Parse(nazivPromenljive);

Menja vrednost tipa podataka string u integer, ako je ta vrednost broj.

float.Parse(nazivPromenljive);

Menja vrednost tipa podataka string u float, ako je ta vrednost broj.

double.Parse(nazivPromenljive);

Menja vrednost tipa podataka string u double, ako je ta vrednost broj.

char.Parse(nazivPromenljive);

Menja vrednost tipa podataka string u char, ako je ta vrednost jedan karakter.

(float)nazivPromenljive;

Menja vrednost tipa podataka integer u float (isto je izvodivo i za double).

(int)nazivPromenljive;

Menja vrednost tipa podataka float ili double u integer.

nazivPromenljive.ToString("#.#");

Menja vrednost decimalnog broja u string i određuje broj decimala (koliko # toliko će se decimala prikazivati).

Convert.ToString(nazivPromenljive);

Menja vrednost promenljive u string.

if

if (uslov) {naredba;}

Ako se ispuni uslov izvršava se naredba, u suprotnom program se nastavlja bez izvršavanja navedene naredbe (isto važi i za blokove naredbi)

if (uslov) {naredba;} else {naredba2;}

Ako se ispuni uslov izvršava se naredba, u suprotnom izvršava se druga navedena naredba (isto važi i za blokove naredbi)

if (uslov) {if (uslov2) {naredba;} else {naredba2;}} else {naredba3;}

If uslov može da bude ugnježđen unutar drugog if uslova što znači da ukoliko prvi uslov nije tačan ide se na izvršavanje naredba3, a ukoliko jeste proverava se uslov2 i u zavisnosti od njegove tačnosti izvršavaju se ili naredba ili naredba2.

if (uslov) {naredba;} else if (uslov2) {naredba2;} else {naredba3;}

U ovom slučaju se nakon provere uslova izvršava naredba ako je uslov tačan, a ako nije proverava se naredni uslov, uslov2, i ukoliko je on tačan izvršava se naredba2, a ukoliko nije naredba3.

switch

switch (nazivPromenljive)

```
{
case vrednost1:
naredba;
break;
case vrednost2:
naredba2;
break;
default :
naredba3;
break;
}
```

Switch se izvršava nad promenljivom, proverava se vrednost i ukoliko jeste identična postavljenoj nakon reči *case* izvršava se naredba a reč *break* zaustavlja izvršavanje svega što je unutar switch bloka nakon toga. Ukoliko vrednost promenljive nije jednak nekoj od ponuđenih vrednosti program izvršava naredbu iz *default* dela. Default deo nije obavezno postaviti.

switch (nazivPromenljive)

```
{
case vrednost1:
case vrednost2:
naredba;
break;
case vrednost3:
naredba2;
break;
default :
naredba3;
break;
}
```

Ukoliko je potrebno proveriti više vrednosti pre izvršavanja neke naredbe provere se pišu jedna ispod druge a naredba (ili blok naredbi) koja treba da se izvrši nakon poslednje provere.

while

while (uslov) {naredba;} Ukoliko je uslov ispunjen naredba će se izvršiti i vratiti se na proveru uslova nakon čega će se ponovo izvršiti ukoliko je uslov i dalje ispunjen a ukoliko nije preskočiće se izvršavanje naredbe.

do while

do {naredba;} while (uslov) Nakon izvršavanja naredbe proveriće se uslov, ukoliko je uslov ispunjen ponovo će se izvršiti ista naredba i ponovo će se proveravati uslov nakon izvršenja sve dok uslov ne prestane biti ispunjen.

for

for (inicijalizacija; uslov; poslednja komanda)

{ naredbe; } Unutar for petlje se prvenstveno upisuje inicijalizacija promenljive, a nakon toga uslov koji se proverava za pomenutu promenljivu i izvršava poslednja komanda pre nove provere uslova.