

Konstante in spremenljivke

Konstante, spremenljivke in prireditveni stavek

Konstante

- `const` – ne spreminjajo svoje vrednosti
 - `const int POVECAJ_ZA = 4;`
 - `const double PI = 3.14;`

```
const double PI = 3.14;  
Console.WriteLine("Obseg kroga s polmerom 5  
cm znaša približno " + 2 * PI * 5 + " cm");
```

Spremenljivke

- Prostor v pomnilniku, kjer hranimo vrednosti
 - Ime
 - Sestavljeno iz znakov in števk
 - Presledkov v imenu ne sme biti!
 - Imena se morajo razlikovati (lokalnost ... a več o tem kasneje)
 - Tip spremenljivke – določa zalogo vrednosti
 - `int` (cela števila), `double` (realna števila), `char` (znak), `string` (zaporedja znakov)
 - Vse spremenljivke napovemo pred prvo uporabo
 - `int sirina;`
 - `int starost, st_cevljev;`
 - Deklaracija, inicializacija, definicija
-

Cela števila: tip int

- Omejen obseg
 - +2 mrd : -2mrd
 - Sami moramo poskrbeti, da so rezultati operacij znotraj tega obsega
 - Operacije: +, -, *, / (celoštevilsko deljenje), % (ostanek pri celoštevilskem deljenju)
 - Prioriteta operacij
 $2 + 3 * 4 \rightarrow 14$
 - Deklaracija + prirejanje začetne vrednosti= definicija
 - `int x;`//deklaracija(napoved, najava spremenljivke)
 - `x=10;`//inicijalizacija(dolocanje zacetne vrednosti)
 - `int x = 10;`//definicija (napoved + inicijalizacija)

V spremenljivki x hranimo cela števila in začetna vrednost je 10.
-

Prireditveni stavek

- V spremenljivko shranimo vrednost
 - `x = 10;`
 - `starost = 25 + 2 * 8;`
 - `ime_spremenljivke = izraz;`
 - Izračuna se vrednost izraza. Dobljena vrednost se shrani v spremenljivko.
 - Če spremenljivka nastopa v izrazu – vrednost, ki jo hranimo v spremenljivki
 - `x = 10; // v x smo shranili 10`
 - `y = 3 * x + 5; // izracunamo izraz: 3 krat število, //ki je shranjeno v x in to povečamo za5 in dobljeni //rezultat shranimo v y.`
 - `x = x + 1; //Vrednost shranjeno v x povečamo za 1`
 - Izračunamo izraz: tisto, kar je shranjeno v x, povečamo za 1.
 - Dobljeni rezultat spet shranimo v x!
-

Realna števila – tip double

- ❑ Realna števila (decimalna števila)
 - ❑ Decimalna pika
 - ❑ Nenatančnost
 - Dvojiški zapis
 - Končno število decimalk
 - 0.1 se ne da napisati točno v dvojiškem sistemu
 - ❑ Operacije: +, -, *, /
 - ❑ Funkcije
 - Razred Math
 - ❑ `double x = 10.2; //DEFINICIJA`

V spremenljivki x hranimo decimalna števila in začetna vrednost je 10.2.
-

Pretvorba realnega v celo število

- Če želimo realno število pretvoriti v celo, napišemo pred realnim številom v okroglih oklepajih *int*.

(int)realnoStevilo

```
double realno = 21.956789;  
int celo = (int)realno; //prevorba realnega števila v celo  
Console.WriteLine(celo); //izpis: 21
```

- Sprememba tipa iz realnega v celo število se opravi tako, da se preprosto odreže decimalni del.
-

Standardne funkcije

- ❑ Elementarne funkcije (logaritem, eksponentna funkcija, sinus, ipd.) so zajete v standardni knjižnici, v razredu **Math**.
 - ❑ Različne metode
 - `Sqrt` (kv. koren)
 - `Sqr` (kvadriranje)
 - `Pow` (Potenciranje)
 - `Abs` (absolutna vrednost)
 - `Sin` (sinus – argument v radianih)
 - ...
 - ❑ Konstanti `Math.PI` in `Math.E` (3.141... in 2.7...)
 - ❑ Funkcije (v C# jim rečemo metode) kličemo npr. z `Math.Sin(izraz)`
 - ❑ `Math.Abs(12 - 15)` je torej 3.
-

Povzetek

- ❑ Vsako spremenljivko je potrebno napovedati (le enkrat!)
 - ❑ Spremenljivko napovemo pred prvo uporabo z deklaracijskim stavkom
`int vsota; // NAPOVED spremenljivke`
 - ❑ Vrednosti prirejamo s prireditvenim stavkom
`// PRIREJANJE vrednosti spremenljivke = inicializacija`
`int vsota=0;`
-

Zgledi

- Spremeni naslednja navodila v ukaze v C#
 - Deklariraj celoštevilčno spremenljivko x z začetno vrednostjo 12.
 - Deklariraj celoštevilčno spremenljivko y z začetno vrednostjo -715.
 - Deklariraj celoštevilčno spremenljivko z z začetno vrednostjo 0.
 - Nastavi z na vsoto spremenljivk x in y.
 - Odštej 7 od x.
 - Nastavi y na produkt x in z.
-

Izpis vrednosti spremenljivke

- ❑ Naj bo x spremenljivka poljubnega tipa, ki že ima neko vrednost
 - `Console.WriteLine(x);`
- ❑ Izpis vrednosti nekega izraza!
 - Vrednost izraza x je vrednost spremenljivke x .
 - `Console.WriteLine(Math.Sqrt(x) + 1);`
 - Izračuna se vrednost izraza.
 - Dobljena vrednost se pretvori v niz
 - V primeru, da je x neko število – zapis tega števila kot zaporedja znakov
 - Izpis tega niza na zaslon.

Zgled: iz števila 38 naredimo 83

□ Algoritem

- Shranimo število
`stevalo = 38;`
- Določimo enice
`enice = stevalo % 10;`
- Določimo desetice
`desetice = stevalo / 10;`
- Naredimo novo število
`novo_stevalo = enice * 10 + desetice;`

□ Program

- Metoda Main
 - Določimo začetno vrednost spremenljivke `stevalo`
 - Nova uporaba: le zamenjamo prireditveni stavek (določimo novo vrednost spremenljivke `stevalo`)
-

Zamenjava vrednosti dveh spremenljivk

- ❑ Pogost opravek v programiranju je menjava vrednosti dveh spremenljivk.
 - ❑ Denimo, da imamo deklarirani dve celoštevilski spremenljivki x in y . Programerji začetniki pogosto mislijo, da se njuni vrednosti zamenja takole:
 - $x = y;$
 - $y = x;$
 - ❑ Kaj se zares zgodi, ko se izvedeta ta ukaza?
 - ❑ Kako se zamenja vrednost dveh spremenljivk?
 - ❑ Namig: uporabi tretjo, pomožno spremenljivko.
-

Zgled: Poraba bencina

- Ko sem se zjutraj peljal v Ljubljano, sem moral natočiti gorivo. Prevozil sem 712 km in natočil 52,2 l goriva. Koliko je bila poraba v l / 100km?
- Spremenljivke
 - `prevozeni_km` : `int`
 - `litri`, `poraba` : `double`
- Izračunamo porabo
$$\text{poraba} = \text{litri} / \text{prevozeni_km} * 100;$$
- Izpišemo rezultat

Program v C#

```
public static void Main(string[] g)//glavni program
{
    string izpis;//niz za oblikovanje izpisa
    double poraba;
    double litri = 52.2;
    int prevozeni_km = 712;

    poraba = litri / prevozeni_km * 100;//izracun porabe
    //oblikovanje izpisa: operator + uporabljamo za stikanje nizov
    izpis = "Ce si prevozil " + prevozeni_km + "km in porabil ";
    izpis = izpis + litri + " goriva,\nje poraba " + poraba;
    izpis = izpis + " l na 100km";

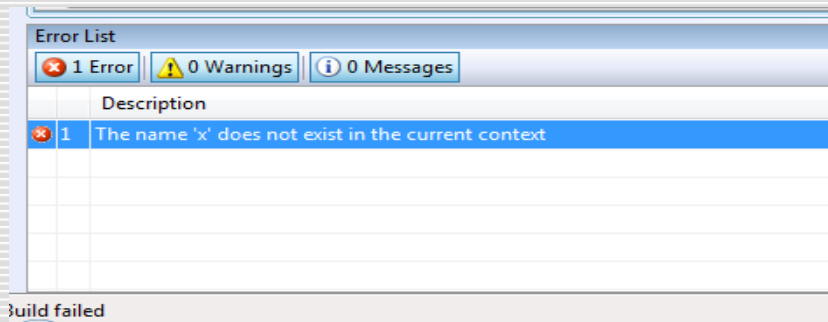
    Console.WriteLine(izpis);
}
```

Napake pri uporabi spremenljivk

□ Pozabimo napovedati spremenljivko

```
static void Main(string[] args)
{
    Console.WriteLine(x);
} // main
```

Obvestilo prevajalnika



Ker spremenljivke *x* pred uporabo nismo najavili, prevajalnik javi napako, saj je ne pozna.

Napake pri uporabi spremenljivk

- Spremenljivki *x* iz prejšnjega programa priredimo novo vrednost 7.

```
static void Main(string[] args)
{
    int x;
    x = 5;
    int x = 7; //NAPAKA: x smo napovedali PONOVRNO!!!
    Console.WriteLine(x);
} // main
```

Spremenljivko v istem bloku napovemo (deklariramo)le enkrat!

Napake pri deljenju z 0

- Paziti moramo, da v programu ne pride do deljenja z nič, saj se v tem primeru izvajanje programa prekine. Izvajalno okolje nam javi, da je prišlo do izjeme (ang. exception).

```
int x = 20;  
int y = 0;  
Console.WriteLine("x = " + x);  
Console.WriteLine("y = " + y + "\n");  
Console.WriteLine("x / y = " + (x / y)); //NAPAKA
```

Napake zaradi prevelikih števil

- Kot smo omenili, lahko v spremenljivkah tipa *int* hranimo cela števila velikosti do približno 2 milijard. Kaj se zgodi, če to mejo prekoračimo?

```
int stevilo1 = 2147483647;
int stevilo2 = 10;
int vsotaStevil = stevilo1 + stevilo2;
Console.WriteLine("2147483647 + 10 = " + vsotaStevil);
```

Dobimo napačen rezultat ($2147483647 + 10 = -2147483639$) brez predhodnega opozorila. Izvajalno okolje ne preverja, če so rezultati izrazov v smiselnih mejah. Na to moramo paziti sami.

- Kadar nam obseg tipa *int* ne zadošča, uporabimo podatkovni tip *long*, a se v podrobnosti ne bomo spuščali.
-

Tabela spremenljivk

□ Sledenje delu programa

Kakšno vrednost imajo spremenljivke a, b in c, ko se izvedejo vsi stavki?

```
int a = 1;
```

```
int b = 2;
```

```
int c;
```

```
c = 3 + 5;
```

```
a = b + 1;
```

```
b = a + c;
```

vrstica	ukaz	a	b	c
1:	int a = 1;	1		
2:	int b = 2;	1	2	
3:	int c;	1	2	
4:		1	2	
5:	c = 3 + 5;	1	2	8
6:	a = b + 1;	3	2	8
7:	b = a + c;	3	11	8

Tabela spremenljivk

Zgled: izpis trimestnega števila navpično

- Trimestno število 376 izpišimo tako, da bodo stotice izpisane v svoji vrstici , nato v svoji vrstici desetice in nato še enice. Programa napišimo tako, da bo za enak izpis nekega drugega trimestnega števila potrebno spremeniti le število 376 v drugo, ostale spremembe v programu pa ne bodo potrebne.
-

Zgled: Trimestno število - obrat

- Napišimo program, ki bo celo število 123 izpisal obrnjeno, torej kot 321. Želimo, da izpis zgloda takole:

Število 123

Obrnjeno število 321
