

Tabele

Enodimenzionalne tabele

Večdimenzionalne tabele

Tabele - uvod

- Dana je naloga: *Preberi 5 števil in jih izpiši v obratnem vrstnem redu.* Vsa števila moramo prebrati (shraniti), ker jih bomo potrebovali in razvrščali šele potem, ko bodo prebrana vsa.

V našem programu moramo ponavljati dva sklopa ukazov. Prvi je

```
Console.WriteLine("Vnesi število:");  
string beri = Console.ReadLine(); //branje vrednosti spremenljivke  
int x = int.Parse(beri);
```

in drugi

```
Console.WriteLine(x);
```

Vsak sklop moramo ponoviti 5x. Razlikujejo se le po tem, da v njih nastopa enkrat x_1 , drugič x_2 , tretjič x_3 ...

Tabele - uvod

□ Program

```
//najprej branje vseh 5 števil
Console.WriteLine("Vnesi število:");string beri=Console.ReadLine();
int x1 = int.Parse(beri);
Console.WriteLine("Vnesi število:");beri = Console.ReadLine();
int x2 = int.Parse(beri);
Console.WriteLine("Vnesi število:");beri = Console.ReadLine();
int x3 = int.Parse(beri);
Console.WriteLine("Vnesi število:");beri = Console.ReadLine();
int x4 = int.Parse(beri);
Console.WriteLine("Vnesi število:");beri = Console.ReadLine();
int x5 = int.Parse(beri);
// izpis
Console.WriteLine("Števila v obratnem vrstnem redu: ");
Console.WriteLine(x5);Console.WriteLine(x4);Console.WriteLine(x3);
Console.WriteLine(x2);Console.WriteLine(x1);
```

Tabele - uvod

- ❑ Namesto običajnih spremenljivk uporabimo raje *tabelarične spremenljivke*.
 - Napoved (deklaracija) tabele

```
podatkovniTip[] imeTabele;
```

Z zgornjim stavkom zgolj napovemo spremenljivko, ki bo hranila **naslov** bodoče tabele, ne zasedemo pa še nobene pomnilniške lokacije, kjer bodo elementi tabele dejansko shranjeni. Potrebno količino zagotovimo in s tem tabelo dokončno pripravimo z ukazom **new**:

- Inicializacija tabele

```
imeTabele = new podatkovniTip[velikost];
```

- Deklaracija in inicializacija (definicija) tabele obenem

```
podatkovniTip[] imeTabele = new podatkovniTip[velikost];
```

Tabele – primeri

- Primeri deklaracije in inicializacije tabele (=definicija tabele)

```
//tabela 10 celih števil: števila dobijo začetno vrednost 0  
int[] stevila = new int[10];
```

```
//tabela 3 realnih števil : števila dobijo začetno vrednost 0  
double[] cena = new double[3];
```

```
//tabela 4 znakov: znaki dobijo začetno vrednost 'prazen znak'  
char[] tabelaZnakov = new char[4];
```

```
//tabela 500 nizov: nizi dobijo začetno vrednost 'prazen niz'  
string[] ime = new string[500];
```

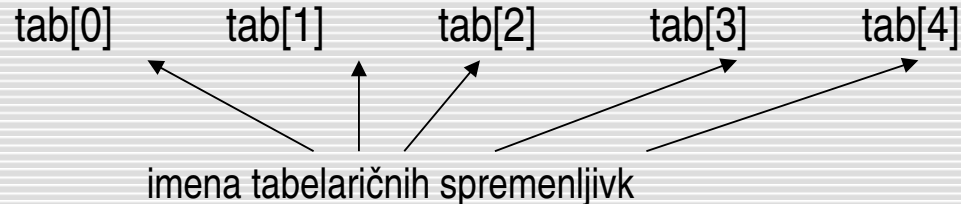
Tabelarične spremenljivke

□ Indeksi

- Ustvarimo tabelo 5 celih števil

```
int[] tab = new int[5]; //ime tabele je tab
```

- Grafično si lahko to tabelo predstavljamo takole:



- Tabelarična spremenljivka

```
tab[4] = 100;
```

Ime tabele Indeks tabelaričnega elementa vrednost tab.spremenljivke

Tabelarične spremenljivke

□ Določanje vrednosti tabelarčnih spremenljivk

- Tabelo najprej ustvarimo

```
//ustvarimo tabelo 5 celih števil: ime tabele naj bo tab  
int[] tab = new int[5]; //vsi elementi so na začetku 0
```

- Tabelaričnim elementom priredimo (oz. spremenimo) ustrezne vrednosti, npr. takole

```
tab[0] = 10;  
tab[1] = 20;  
tab[2] = 31;  
tab[3] = 74;  
tab[4] = 97;
```

- Ali pa npr. takole (vrednosti vnese uporabnik)

```
for (int i = 0; i <= 4; i++)  
{  
    Console.WriteLine("Vnesi število:");  
    string beri = Console.ReadLine();  
    x[i] = Convert.ToInt32(beri);  
}
```

Tabele

- Izpis vrednosti elementov, ki so v tabeli: obravnavati moramo vsakega posebej.

```
/*definicija tabele n elementov*/  
podatkovniTip[] imeTabele = new podatkovniTip[n];  
  
//velikost tabele določa lastnost Length  
for (int i = 0; i < imeTabele.Length; i++)  
{  
    //izpis tabelaričnega elementa z indeksom i  
    Console.Write(imeTabele[i]);  
}
```

Zgled – tabela celih števil

- ❑ Preberi 5 števil in jih izpiši v obratnem vrstnem redu.

```
//branje
string beri;
int[] x = new int[5]; //deklaracija tabele 5 celih števil
for (int i = 0; i <= 4; i++)
{
    Console.WriteLine("Vnesi število:");
    beri = Console.ReadLine(); //preberemo vhodni podatek
    /*vneseno vrednost spremenimo v celo število in jo
    shranimo v tabelarično spremenljivko*/
    x[i] = int.Parse(beri); //ali Convert.ToInt32(beri)
}
//tabelo izpišemo v obratnem vrstnem redu
for (int i = 4; i >= 0; i--)
    Console.WriteLine(x[i]);
Console.ReadKey();
```

Zgled – tabela naključnih števil

- ❑ Ustvarimo tabelo 100 celih števil in jo inicializirajmo tako, da bodo imeli vsi elementi tabele vrednost naključnega celega števila med 0 in 10.

```
int[] tabela = new int[100]; // definicija tabele
Random naklj = new Random(); // generator naključnih števil
for (int i = 0; i < 100; i++)
{
    // naključne vrednosti med 0 in 10
    tabela[i] = naklj.Next(11);
}
```

Tabele - velikost

- ❑ Velikost tabele lahko določimo tudi med samim delovanjem programa. Dimenzijo tabele torej lahko določi uporabnik, glede na svoje potrebe

```
Console.Write("Določi dimenzijo tabele: ");
```

```
// Dimenzijo tabele določi uporabnik med izvajanjem!!!  
int dimenzija = int.Parse(Console.ReadLine());
```

```
int[] tabela = new int[dimenzija];
```

Tabele – inicializacija

- Začetno vrednost tabelarnih spremenljivk lahko določimo tudi takole (tak način ima smisel le če je tabelarnih spremenljivk malo)

```
/*deklaracija in inicializacija tabele 9 celih števil  
   elementi tabele dobijo vrednost od 1 do 9*/
```

```
int[] stevila = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };
```

Tabele – podobnost z nizi

- ❑ Izraz, s katerim dostopamo do i -tega elementa tabele ($tab[i]$), se po videzu ujema z izrazom za dostopanje do i -tega znaka v nizu ($niz[i]$). **Izraza pa se ne ujemata v uporabi.** Pri tabelah uporabljamo izraz za pridobitev in spremembo elementa tabele, medtem ko ga pri nizih uporabljamo le za pridobitev znaka. V primeru, da izraz uporabimo za spremembo določenega znaka v nizu, nam prevajalnik vrne napako.

```
int[] tab = { 1, 6, 40, 15 }; //Inicializacija tabele
string niz = "Trubarjevo leto 2008."; //Inicializacija niza
// Spreminjanje
tab[1] = 5; // Spremenjena tabela je [1, 5, 40, 15].
niz[18] = '9';/*NAPAKA! Znaka v nizu ne moremo spremeniti na
    ta način.*/
```

Tabele – najpogostejše napake

- ❑ Napačna deklaracija in inicializacija tabele

```
string[] stevila = { 1, 2, 3 };  
Console.WriteLine("Dolžina tabele: " + stevila.Length);
```

Program se ne prevede. Deklarirali smo tabelo nizov, tej tabeli pa smo priredili celoštevilске vrednosti. Ker se tipa ne ujemata, bo prevajalnik javil:

Error 3 Cannot implicitly convert type 'int' to 'string'

Napako popravimo tako, da napišemo ustrezen (pravilen) tip.

```
int[] stevila = { 1, 2, 3 };  
Console.WriteLine("Dolžina tabele: " + stevila.Length);
```

Tabele – najpogostejše napake

□ Primerjanje dveh tabel

- dani sta dve tabeli *t1* in *t2*:

```
int[] t1 = new int[] { 1, 2, 3 };  
int[] t2 = new int[] { 1, 2, 3 };
```

- primerjajmo ju, ali sta enaki

```
Console.WriteLine(t1 == t2);
```

Izpis (*false*) pove, da tabeli nista enaki. Zakaj? Spomnimo se, da *t1* in *t2* vsebujeta samo naslov, kjer se nahajata tabeli. Ker sta to dve različni tabeli (sicer z enakimi elementi, a vseeno gre za dve tabeli), zato tudi naslova **nista enaka**. In to nam pove primerjava `t1 == t2`.

Enakost dveh tabel lahko ugotavljamo le tako, da najprej primerjamo njuni dimenziji, nato pa še vsak element posebej!

Zgled: Število besed v stavku

- ❑ Preberi poljuben stavek. Koliko besed vsebuje?

```
// Vnos niza
Console.Write("Vnesi niz: ");
string niz = Console.ReadLine();
// Pretvorba niza v tabelo nizov
string[] tab = niz.Split(' ');
// Iskanje najdaljše besede
string pomocni = ""; // Najdaljša beseda (oz. podniz)
for (int i = 0; i < tab.Length; i++)
{
    if (pomozni.Length < tab[i].Length)
    {
        pomocni = tab[i];
    }
}
// Izpis najdaljše besede
Console.WriteLine("Najdaljša beseda: " + pomocni);
```



Vaje

- ❑ Smiselno dopolni naslednji program

```
string[] _____ = {"Jan", "Ana", "Tim", "Maja", "Tine"};  
Console.WriteLine("Dolžina tabele je " + _____);  
    Console.WriteLine("Prvo ime v tabeli je " + imena[0]);
```

- ❑ Predpostavimo, da imamo tabelo z naslednjimi elementi: 2, 5, 7, 1, 6, 10, 3, 8, 0 in 11. Napišimo program, ki bo določil in izpisal vsoto vseh elementov in njihovo povprečno vrednost
 - ❑ Napiši program, ki prebere določeno število nizov in jih izpiše v obratnem vrstnem redu. Podatek o tem, koliko nizov bo vnesenih, prebereš na začetku programa.
-

Vaje

- ❑ Ustvari poljubno tabelo nizov in jo uredi po abecedi!

```
string[] tab = { "Kranj", "Ljubljana", "Celje",  
                "Ankaran" };
```

```
for (int i = 0; i < tab.Length; i++)  
    Console.Write(tab[i] + " ");
```

```
Console.WriteLine();
```

```
Array.Sort(tab); //tabelo uredimo po abecedi!
```

```
for (int i = 0; i < tab.Length; i++)  
    Console.Write(tab[i] + " ");
```

Vaje

- Dana sta ukaza

```
string[] stavek = new string[10];  
stavek[0] = "Konec se bliža";
```

Kateri ukaz moramo uporabiti, če želimo izpisati dolžino niza "Konec se bliža"?
Obkroži vse pravilne odgovore!

- Console.WriteLine(stavek.Length());
 - Console.WriteLine(stavek[0].Length);
 - Console.WriteLine(stavek.Length);
 - Console.WriteLine(stavek[0].Length());
-

Vaja

- ❑ Napiši program, ki bo izpisal enodimenzionalno tabelo naključnih celih števil med 0 in 100 (dimenzijo tabele izbere uporabnik programa), poiskal njen največji element ter izračunal, za koliko se največji element razlikuje od povprečne vrednosti vseh elementov.
 - ❑ Ustvari naključno tabelo 100 celih števil z vrednostmi med 10 in 100 in izpiši le tiste člene, ki so večji od zadnjega elementa v tej tabeli. Program dopolni tako, da ustvariš naključno dolgo tabelo (a ne krajšo od 10 in ne daljšo od 100 elementov).
-