

POLJA

1

KO ZMANJKA SPREMENLJIVK...

Kaj so polja?

2

- **polje** je zbirka elementov istega podatkovnega tipa
 - zbirka imen
 - zbirka povprečne letne temperature v zadnjih 50 letih
- **Zakaj potrebujemo polja?**
 - zamislite si zgoraj omenjeno zbirko povprečnih letnih temperatur za zadnjih 50 let (ali 100, ali celo 500)
 - ✦ kako bi poimenovali vse te spremenljivke?
 - ✦ kako bi obdelali vse te spremenljivke?

Deklaracija polja - primer

3

- polje, poimenujmo ga *ocene*, ki vsebuje recimo 5 celih števil, deklariramo na naslednji način:

```
int ocene[5];
```

- ta deklaracija je enaka, kot če bi deklarirali pet celih števil z imeni:

```
ocene[0], ocene[1], ..., ocene[4]
```

Deklaracija polja - splošno

4

- v splošnem polje z n elementi podatkovnega tipa *tip*, deklariramo na naslednji način:

```
tip ime[n];
```

- *tip* ... podatkovni tip elementov v polju
- *ime* ... ime polja (ime zbirke podatkov)
- *n* ... število elementov v polju (mora biti konstanta)

Polja

5

- spremenljivke, ki sestavljajo polje, imenujemo **elementi polja**
- število elementov v polju imenujemo **velikost polja**
- položaj elementa v polju imenujemo **indeks polja**
 - prvi element je na indeksu 0
 - zadnji element je na indeksu velikost-1

Polja

6

- elementi polja so lahko poljubnega podatkovnega tipa
- vsi elementi polja imajo
 - enak podatkovni tip
 - natančno določen položaj v polju – indeks
- elemente polja lahko uporabljamo kot običajne spremenljivke
- do posameznega elementa v polju lahko dostopamo preko operatorja [] in indeksa elementa

Polja – operator []

7

- do posameznega elementa v polju dostopamo preko
 - imena polja
 - indeksa elementa, ki ga želimo

- npr. do 5. elementa v polju z imenom *podatki* dostopimo tako:

`podatki[4]`

- v splošnem: `imePolja[indeksElementa]`

Polja – operator []

8

- sami moramo paziti na to, da element z zapisanim indeksom obstaja
 - npr. ne moremo dostopati do 15. elementa, če je v polju samo 7 elementov
- vse, kar delamo s poljem, lahko delamo le s posameznimi elementi

PRIMER:

```
int podatki[10]; // polje desetih celih števil
```

```
...
```

```
cout << "Vnesi podatke v polje: ";
```

```
cin >> podatki; // NAROBE!!!!
```

Polja in zanke

9

- če želimo kaj storiti s celotnim poljem, moramo to storiti z vsemi elementi polja (uporabimo zanke!)

PRIMER:

```
int podatki[10]; // polje desetih celih števil
cout << "Vnesi podatke v polje: ";
for(int i=0; i<10; i++) {
    cout << "Vnesi " << i << ". element: ";
    cin >> podatki[i];
}
```

Polja in zanke

10

- če želimo kaj storiti s celotnim poljem, moramo to storiti z vsemi elementi polja (uporabimo zanke!)

PRIMER:

indeks prvega elementa je 0

indeks zadnjega elementa je 9 (torej <10)

```
cout << "Vnesi podatke v polje: ";
for(int i=0; i<10; i++) {
    cout << "Vnesi " << i << ". element: ";
    cin >> podatki[i];
}
```

Inicializacija polja

11

- elemente polja lahko tudi inicializiramo ob deklaraciji polja
 - vsakemu elementu določimo začetno vrednost
 - vrednosti elementov zapišemo znotraj { } ločene z vejicami

PRIMER:

```
int ocene[3] = {7, 8, 8};
```

je enako kot

```
int ocene[3];
ocene[0] = 7;
ocene[1] = 8;
ocene[2] = 8;
```

Inicializacija polja

12

- če ob inicializaciji polja podamo manj vrednosti, kot je elementov, se vsi preostali elementi inicializirajo na vrednost 0

PRIMER:

```
int podatki[10] = {7, 8, 8};
```

- prvi tri elementi so 7, 8 in 8, ostalih sedem elementov je enakih 0

Uporaba konstant za velikost polja

13

- velikost polja mora biti konstanta
- konstanto deklariramo kot spremenljivko, pri kateri pred deklaracijo napišemo **const**

```
const int VEL = 15;
int podatki[VEL];
...
cout << "Izpis elementov polja: ";
for (int i=0; i<VEL; i++)
    cout << podatki[i] << " ";
```

Polja - naloge

14

1. Napišite program, ki bo v ustvaril polje 10 celih števil in ga napolnil z naključnimi lihimi števili med 1 in 1000. Nato izračunajte in izpišite na ekran vsoto in povprečno vrednost vseh elementov v polju.
2. Napišite program, ki preko tipkovnice bere realna pozitivna števila v polje tako dolgo, dokler ne zapolnimo celotnega polja (10 elementov). Izpišite najmanjši in največji element v polju.
3. Napišite program, ki ustvari polje z 10 elementi, ki vsebuje naključna števila med 1 in 10. Nato preverite, ali so vsi elementi v polju soda števila.