ALGEBRA

PROGRAMIRANJE

Predavanje 06 – Vektori

Ishod učenja 2

1

Glavni problem s poljima

- Glavni problem s poljima: moramo odlučiti o veličini polja u trenutku deklariranja
  - U realnim situacijama skoro nikada to ne možemo znati

ALGEBRA

2

Glavni problem s poljima

- Primjerice, kako možemo korisniku omogućiti da unese onoliko brojeva koliko želi
  - Loša opcija 1: deklarirati polje veličine 1000 i nadati se da korisnik neće unijeti toliko
    - Zašto je ovo loše:
      - Rezerviramo 1000*4 bajtova čak i ako korisnik unese samo 1 broj
      - Uporni korisnik ipak može probiti limit
  - Loša opcija 2: deklarirati polje veličine 50 i spriječiti korisnika da unosi dalje
    - Zašto je ovo loše:
      - Rezerviramo 50*4 bajtova čak i ako korisnik unese samo 1 broj
      - Korisniku može trebati više od 50

ALGEBRA

3

Vektor

- Postoje dva dobra rješenja:
  - Izravno korištenje dinamičkih polja (u ishodu učenja 5)
  - Novi tip podataka: vektor
- **Vektor** možemo smatrati pametnim poljem koje je prazno u trenutku deklariranja i koje raste prema potrebi
- Definiran u zaglavlju `vector`:

```
#include <vector>
```

- Pri deklariranju, koristimo razlomljene zagrade da definiramo tip podataka elemenata koje će sadržavati, npr:

```
vector<string> students;
vector<int> numbers;
```


ALGEBRA

4

Upotreba vektora

- Novi elementi se dodaju u vektor pozivanjem metode **`push_back()`**:

```
students.push_back("Miro");
numbers.push_back(71);
```

- Ubačenim elementima pristupamo koristeći indekse, kao kod običnih polja:

```
cout << students[0];
numbers[0]++;
```

- Broj elemenata dobijemo pozivanjem metode **`size()`**
  - `size()` vraća broj elemenata, a ne broj bajtova

Primjer korištenja vektora

- Primjer korištenja vektora:

```
vector<int> numbers;
int n;
cin >> n;
numbers.push_back(n);
cout << numbers.size() << ", " << numbers[0] << endl;
```


Primjeri

1. Učitavajte od korisnika imena sve dok ne kaže da ne želi unositi više imena. Ispišite učitane stringove obrnutim redoslijedom.
2. Učitavajte od korisnika brojeve sve dok ne kaže da ne želi unositi više brojeva. Ispišite količinu i sumu unesenih brojeva.
3. Neka je zadano polje od 9 cijelih brojeva s vrijednostima: 11, 22, 33, 44, 55, 66, 77, 88, 99. Prepišite brojeve iz polja u vektor, ali obrnutim redoslijedom. Ispišite brojeve iz vektora.

ALGEBRA

7

Primjeri

4. Učitavajte od korisnika brojeve dok on to želi. Ispišite najmanji učitani broj.
5. Upišite cijele brojeve od 2 do 10.000 u polje. Prepišite sve proste brojeve iz polja u vektor. Ispišite brojeve iz vektora.
6. Riješiti sve zadatke iz predavanja o poljima, ali koristeći vektor

ALGEBRA

8

Zadaci za sljedećih 7 dana

- Prije sljedećeg predavanja trebate:

1. Pročitati iz *Demistificirani C++*:
   - 8.1 Što su i zašto koristiti funkcije
   - 8.2 (without 8.2.1) Deklaracija i definicija funkcije
   - 8.3 (without 8.3.1) Tip funkcije
   - 8.4.1 Funkcije bez argumenata
   - 8.4.2 Prijenos argumenata po vrijednosti

9

Zadaci za sljedećih 7 dana

2. Pogledati sljedeće:
   - W07-1 Functions 1
     - https://youtu.be/mWTvyKEOSUU
   - W07-2 Functions 2
     - https://youtu.be/zYB2BnUPmGU
   - W07-3 Solving problems
     - https://youtu.be/8jRNcAYeQXo

10