

ZADAĆA 3

Objektno orijentirano programiranje

Upute:

Zadaću je potrebno predati do 20. ožujka u 08:00 na Teamsu. Diskutiranje zadaće u okviru “study group” je dozvoljeno, ali predana zadaća mora biti samostalno riješena. Studenti koji predaju zadaću obavezni su prisustvovati vježbama, u suprotnome zadaća neće biti bodovana. Ako se kod ne prevodi (neovisno o tipu compiler-a), ili se događa greška prilikom izvršavanja koda, zadaća neće biti bodovana.

Zadatak 1. (35 bodova) Polje radijvektora

Klasa `PositionVector` dana je sljedećom deklaracijom:

```
struct PositionVector {  
    double x, y;  
  
    PositionVector();  
    PositionVector(double, double);  
    PositionVector(const PositionVector&);  
  
    double norm() const;  
    void print() const;  
};
```

Implementirajte odgovarajuće članove. Prilikom ispisa instance ove klase ispišite kordinatu x, kordinatu y te normu.

Statički alocirajte polje 5 radijvektora s koordinatama redom (1.5, 2.7), (3.4, 5.1), (2.3, 6.3), (5.0, 0.8) i (0.0, 0.0). Ispišite redom elemente tog polja koristeći range-based for petlju. Sortirajte polje radijvektora s obzirom na normu vektora u padajućem poretku. Ponovno ispišite redom elemente polja.

Zadatak 2. (45 bodova) Bodovi na ispitu

Student Marko želi izbrojati koliko je drugih studenata riješilo ispit bolje ili poput njega. Rezultati pismenog ispita nalaze se u datoteci `data.txt` (dostupna u zadaći) koja se treba nalaziti u istoj mapi kao i source kod ovog zadatka. U toj datoteci u prvom se retku nalazi broj studenata koliko ih je pristupilo ispitu. U ostalim se retcima nalaze bodovi po zadacima za svakog od studenata pri čemu je u prvom takvom retku broj bodova od Marka, a u ostalim retcima bodovi ostalih studenata.

Npr. sadržaj

```
5
30 35 25 15 10
20 30 40 15 10
30 30 30 10 10
45 25 30 10 10
25 35 20 15 10
```

u `data.txt` datoteci broj 5 znači da je pet studenata pristupilo ispitu, Markov rezultat po zadacima je 30, 35, 25, 15, 10 bodova, a ostalih studenata kako slijedi u ostalim redcima.

Definirajte članove klase `Student` čija je deklaracija

```
struct Student {
    int scores[5];
    Student();
    void fillScores(int(& arr)[5]);
    int calculateTotalScore() const;
    void print() const;
};
```

pri čemu `scores` predstavlja polje ocjena po zadacima za nekog studenata, a elementi tog polja se popunjavaju pomoću `fillScores`. Koristeći `std::accumulate` definirajte `calculateTotalScore` koja zbraja sve elemente polja `scores`, a `print` ispisuje elemente polja.

Parsirajte datoteku `data.txt`, dinamički alocirajte polje instanci klase `Student` odgovarajuće veličine, popunite ocjene za svakog studenta te koristeći `count_if` prebrojite koliko njih ima veći ili jednak broj bodova kao Marko. Npr., za sadržaj

```
5
30 35 25 15 10
20 30 40 15 10
30 30 30 10 10
45 25 30 10 10
25 35 20 15 10
```

u `main` -u se treba ispisati

2

Napomena. Implementacija treba raditi za proizvoljan slučaj s n studenata.

Zadatak 3. (20 bodova) Iterabilnost klase

U drugoj je zadaći vaš zadatak bio implementirati dinamičko polje kao klasu `MyVector`. Nadogradite ovu klasu metodama `begin` i `end` kako bi instanca ove klase bila iterabilna, odnosno, kako bismo bili u mogućnosti iterirati ju kroz range-based `for` petlju. U main dijelu kreirajte barem jedan objekt klase `MyVector` i iterirajte kroz njega koristeći range-based `for` petlju.

Napomena: u klasu je potrebno dodati sljedeću deklaraciju za metode `begin` i `end`

```
/*  
početak i kraj vektora  
(obratite pažnju što treba vraćati end)  
*/  
int* begin() const;  
int* end() const;
```