

## PROGRAMSKI JEZICI 1 - K1

- ❶ (25 bodova) U prostoru imena **kolokvijum** definisati klasu **Tacka** koja predstavlja tačku u dvodimenzionom (kontinualnom) koordinatnom sistemu. Tačku karakterišu koordinate i naziv.

Potrebno je implementirati dva konstruktora: prvi, koji prihvata naziv kao **char\*** i koordinate i drugi, koji prihvata samo koordinate, sa podrazumijevanim vrijednostima.

U prostoru imena **kolokvijum** treba da postoji konstantan objekat klase **Tacka** koji predstavlja koordinatni početak. Tačke čiji naziv nije zadat treba da imaju prazan tekst naziva.

Pored toga, klasa **Tacka** treba da ima:

- Operatore poređenja (jednakost i razliku) koji porede dvije tačke samo po koordinatama
- **Operator** - koji nalazi rastojanje između dvije tačke
- Odgovarajuće setere i getere za koordinate (seter koordinata realizovati jednom funkcijom)
- Operator kastovanja u **double** koji vraća rastojanje tačke od koordinatnog početka
- Operatore za ispis i učitavanje tačke (ispis treba da bude formatiran, a učitavanje u obliku: *naziv x y*)

- ❷ (30 bodova) U prostoru imena **kolokvijum** definisati klasu **Niz** koja predstavlja niz tačaka (objekata klase **Tacka**). Niz treba da bude dinamički alociran, fiksne dužine proslijeđene kroz konstruktor (broj elemenata mora biti bar jedan). Potrebno je realizovati konstruktor kopije, destruktor i operator dodjele. Osim toga, potrebno je:

- Realizovati **operatore** **+=**, **-=** i **\*=** tako da prihvataju vrijednost tipa **double** i mijenjaju sve koordinate svih tačaka niza za tu vrijednost u skladu sa osnovnom funkcionalnošću svakog od navedenih operatora
- Realizovati operator indeksiranja tako da vraća element na zadanom indeksu. Ukoliko je indeks van opsega niza, vratiti nulti element.
- Realizovati metodu **filtriraj(const std::function<bool(const Tacka\*)>&)** koja vraća novi niz elemenata (objekat klase **Niz**) koji sadrži kopije onih tačaka niza koje su zadovoljile uslov proslijeđen metodi.

- ❸ (20 bodova) U glavnom programu instancirati tri tačke: **a**, bez imena sa koordinatama (0,0), **b**, sa imenom „B” i koordinatama (2,4) i **c**, sa imenom „C” i koordinatama (1,2). Potom, instancirati objekat klase **Niz** sa četiri elementa i redom elementima niza dodijeliti tačke **a**, **b** i **c**. Četvrtu tačku niza učitati sa standardnog ulaza. Potom:

- Ispisati niz na standardni izlaz
- Na standardni izlaz ispisati samo one tačke niza čije su koordinate jednake koordinatama četvrte tačke u nizu
- Učitati četiri tačke sa standardnog ulaza i za svaku tačku redom provjeriti da li odgovarajuća tačka iz niza ima te koordinate, ako joj se koordinate skaliraju dva puta; u slučaju da to nije tačno, ispisati datu tačku na standardni izlaz.

Dat je tekst u konzolnom prozoru nakon izvršavanja ovog programa:

Ulaz: T 1 2

(0, 0)

B(2, 4)

C(1, 2)

T(1, 2)

C(1, 2)

T(1, 2)

Ulaz: A 0 0

Ulaz: A 4 8

Ulaz: A 10 10

Ulaz: A 4 8

C(2, 4)

T(2, 4)

- ❹ (25 bodova) Odrediti izlaz iz sljedećeg programa:

```
#include <iostream>
#include <functional>
using namespace std;
struct A {
    A(int a) : a(a) { cout << "cA" << a << ' '; }
    A(const A& t) { cout << "xA" << (a = t.a) << ' '; }
    ~A() { cout << ' ' << "dA" << a; }
    int a;
};
class B {
public:
    B(function<void(A&> x) : f(x), b(1), a(2)
    { cout << "B" << ' '; }
    void fun() { f(a); }
private:
    A a, b;
    function<void(A&> f;
};
A w(3);
int main() {
    A q(4);
    B b[](A& a) { cout << "f" << a.a; });
    b.fun();
}
```

### Napomene:

Razdvojiti interfejs i implementaciju.

Izbjeći dupliciranje koda.

U implementaciji klasa koristiti punu specifikaciju imena objekata i metoda iz standardne biblioteke.

Koristiti standardnu biblioteku.