ZADAĆA 2: OSNOVE KLASA

Svaki zadatak nosi određeni broj bodova po podzadatku koji su naznačeni u zagradi. Prvi zadatak treba biti predan kao txt dokument (napravljen u SC5), a ostali kao C++ programski kod. Svi student koji predaju zadaće trebaju pristupiti na ScriptRunner5 sustav (https://mathos.scriptrunner.carnet.hr/) I tamo objavljivati svoje zadaće u mapi ZADACE->ZADACA_2.

Svi oni studenti koji predaju zadaće moraju biti nazočni na vježbama.

ZADATAK 1 (5+5+5):

Sljedeća pitanja se tiču razumijevanja upotreba klasa u C++. Za svaki odgovor dajte objašnjenje zašto je točno, odnosno netočno.

1. Što je problem u sljedećem programskom kodu i kako bi ga popravili da radi? Odgovor priložite.

```
#include <iostream>
using nanespace std;
class Klasa
{
    public:
        Klasa A(){x=2;y=1;}

    protected:
        void test();

        int x, y;


};

void test()
{
   cout <<  x << " " << y << endl;
}

int main()
{
    Klasa A(2,1);
    Klasa B;

    cout <<  A.x << " " << A.y << endl;
    B.test();
}
```

```
// korekcija programskog koda
```

2. Koji od sljedećih naredbi je valjana u C++-u i dajte obrazloženje svojim odgovorima:

   ☐ `void Klasa::test();`

   ☐ `int Klasa::void();`

   ☐ `int Klasa.test();`

ZADATAK 2 (10+5+5):

U programskom jeziku C++ postoji implementacija dinamičkog polja `vector<T>`. Implementirajte svoju verziju vektora koja koristi sljedeći potpis:

```
#include <iostream>
using namespace std;
class vector
{
    public:
      vector(){}
      vector(size_t size);
      size_t size() const; //size_t = unsigned int
      size_t capacity() const;

      // dodavanje i brisanje elemenata s kraja
      void push_back(float x);
      float pop_back();
      void resize_to_fit();
      void clear();
};
```

Klasa podržava skidanje i vraćanje elementa s kraja metodom `pop_back`, dodavanje elementa x na kraj polja s `push_back` i brisanje svih elemenata iz polja s `clear`. Veličina polja se dinamički povećava odnosno smanjuje u ovisnosti o broju elemenata `size`. Ukoliko `push_back` dodaje element u već puno polje `(size==capacity)` onda se polje povećava na polje duljine `capacity*2`. Ukoliko `pop_back` operacija briše element za koje je `size==capacity/2` onda se polje smanjuje na veličinu `capacity/2`. Metodom `resize_to_fit` postavlja se kapacitet polja na `size`.

ZADATAK 3 (30+10)

Neka je dana klasa `Tocka` koja implementira objekt koji predstavlja točku unutar euklidske ravnine.

```
class Tocka
{
    public:

      float x;
      float y;

      Tocka(){};
      Tocka(float x, float y){};
      Tocka(const Tocka& T){};


};
```

1. Definirajte klasu `Pravokutnik` koja koristi klasu `Tocka` da implementira objekt `Pravokutnik` i pripadne metode za računanje površine i opsega:

```
class Pravokutnik
{
    public:
     Pravokutnik(){};
     Pravokutnik(const Tocka&, const Tocka& ){};
     Pravokutnik(const Pravokutnik& ){};

     // metode za racunanje
     float opseg() const{};
     float povrsina() const{};

     Tocka T1,T2;
};
```

2. Implementirajte klasu `PravilniPoligon` koja je dana sljedećim potpisom:

```
class PravilniPoligon:
{
    public:
   PravilniPoligon(){};
   PravilniPoligon(const Tocka& S, const Tocka& T, int n){};
   PravilniPoligon(const PravilniPoligon& P){};

      float opseg() const{};
      float povrsina() const{};

      Tocka S, T1;
      int n;
      Tocka *vrhovi;
};
```

gdje S označava središte pravilnog poligona, a T prvi vrh. Koristeći središnji kut poligona, konstruirajte preostalih $n - 1$ točaka u konstruktoru. Za konstrukciju koristite elementarno poznavanje trigonometrije, a točke spremite interno u polje vrhovi. Dodatno, implementirajte metode koje će vratiti površinu i opseg mnogokuta.

Implementaciju metoda definirajte izvan klasa. Oba zadatka testirajte sljedećim primjerom:

```
int main()
{
    Pravokutnik P(Tocka(1,3), Tocka(2,5));
    cout << "***P: opseg:" << P.opseg() << ", povrsina: " << P.povrsina() << endl;

  PravilniPoligon P_5(Tocka(1,5), Tocka(3,6), 5);
    cout << "***P_5: opseg:" << P_5.opseg() << ", povrsina: " << P_5.povrsina() << endl;

}
```