

### Zadanie 1 (4p)

---

Napisz klasę wektor, oraz konstruktory klasy:

- a) domyślny, przyjmujące dwa obiekty typu pkt, określające punkty początku i końca wektora.

A także:

- a) destruktor,
- b) funkcje składową, która służą do dodawania i odejmowania wektorów,
- c) mnożenie skalarne,
- d) funkcję składową, która zwraca ilość stworzonych dotychczas obiektów typu wektor.

Stwórz obiekt tej klasy i wykonaj operacje mnożenia, dodawania, odejmowania wektorów oraz wypisz ile elementów obiektu wektor zostało stworzonych.

Ile miejsca w pamięci zajmuje obiekt klasy wektor?

### Zadanie 2 (3p)

---

Zdefiniuj klasę koło, która powinna zawierać:

poła prywatne:

- a) promień koła (typ float)
- b) kolor wypełnienia (typ int)
- c) konstruktor bezparametrowy inicjujący wartość początkową pola koła (dowolnie ustaloną)
- d) konstruktor z parametrami promień, kolor wypełnienia

metody publiczne:

- a) Oblicz\_Pole(..), obliczająca pole powierzchni koła
- b) Ustaw\_Promien(..), zmieniająca promień koła
- c) Ustaw\_Kolor(..), zmieniającą kolor koła
- d) funkcję zaprzyjaźnioną Porównaj\_Kola(KOLO A, KOLO B), porównującą, czy dwa koła mają taką samą powierzchnię i kolor.

Uruchom w programie głównym sekwencję instrukcji dla dwóch obiektów K1, K2: konstruktor bezparametrowy, porównanie pól, ustawienie tego samego koloru, porównanie pól, ustawienie różnych kolorów, porównanie pól.

### Zadanie 3 (3p)

---

Utwórz klasę punkt. Klasa powinna zawierać:

- a) 2 pola prywatne typu całkowitego - współrzędne punktu
- b) konstruktor bezparametrowy inicjujący dowolne wartości początkowe punktu

- c) konstruktor z parametrami - współrzędne punktu
- d) metodę publiczną - zwracającą współrzędną x
- e) metodę publiczną - zwracającą współrzędną y

Zdefiniuj funkcję zaprzyjaźnioną porównującą dwa punkty, czy się nakładają. Uruchom program dla dwóch różnych obiektów, porównaj je, a następnie dokonaj zmian w jednym z nich, by uczynić obiekty identycznymi. Spróbuj wykonać te same działania w wersji z dynamicznym przydziałem pamięci.

#### Zadanie 4 (2p)

---

Czy któreś z poniższych stwierdzeń jest fałszywe? Jeżeli tak, to które? Dlaczego tak sądzisz?

- a) Klasa musi zawierać co najmniej jeden konstruktor.
- b) Lista argumentów formalnych każdego konstruktora domyślnego jest pusta.
- c) Jeżeli nie można określić sensownych wartości domyślnych dla klasy, to nie można definiować dla tej klasy konstruktora domyślnego.
- d) Jeśli klasa nie zawiera jawnego konstruktora domyślnego, to kompilator automatycznie generuje konstruktor, który inicjuje każde pole klasy wartością domyślną właściwą dla typu pola.

#### Zadanie 5 (3p)

---

Wybierz jedno z poniższych pojęć abstrakcyjnych (albo wymyśl własne). Po określeniu danych odpowiednich do reprezentowania pojęcia (u ustawiania przez użytkowników) zdefiniuj własny zbiór konstruktorów.

- a) Książka
- b) Data
- c) Pracownik
- d) Pojazd
- e) Obiekt
- f) Drzewo

#### Zadanie 6 (2p)

---

Korzystając z następującej definicji klasy Rachunek :

```
class Rachunek {
public:
    Rachunek();
    explicite Rachunek(const char*, double=0.0);
```

```
        //...  
};
```

wyjaśnij znaczenie poniższych definicji:

- Rachunek rach;
- Rachunek rach3 = "Rena Stern";
- Rachunek rach4(„Anna Engel”,400.00);

### Zadanie 7 (2p)

---

Napisz definicję destruktora odpowiednią dla poniższego zbioru pól klasy, wśród których pole napisu wskazuje na dynamiczną tablicę znaków.

```
class BezNazwy {  
    public:  
        ~BezNazwy();  
        //...  
  
    private:  
        char *wnapisu;  
        int wrti;  
        double wrtd;  
};
```

### Zadanie 8 (1p)

---

Jak myślisz, dlaczego język C++ nie nazywa się ++C?

### Zadanie 9 (2p)

---

Stwórz

- tablicę 10 obiektów klasy pojazd
- wskaźnik do klasy pojazd
- funkcję zaprzyjaźnioną z klasą pojazd - zatrzym\_sie()

Max 22p

**Oceny :**

**0-8p ndst.**

**8,5-11,5p dop**

**12-15p dst**

**15,5-18,5p db**

**18,5p+ bdb**