

Laboratorium 2

Klasy

Załóżmy, że program zawiera poniższą definicję klasy

```
class T
{
    int liczba;
    char nazwa;
public:
    T();
    T(int nowa_liczba, char nowa_nazwa);
    void drukuj();
};
```

Które z poniższych wyrażeń są poprawne?

```
T obiekt(1, 'A');
T inny_obiekt;
T jeszcze_inny_obiekt();
obiekt=T(2, 'B');
obiekt=T();
obiekt=T;
```

Zadanie 2.

Dana jest deklaracja klasy Punkt. Napisz definicję funkcji oraz program, który ilustruje działanie klasy:

```
class Punkt
{
    double x,y;           // współrzędne kartezjańskie
public:
    Punkt(double,double); // konstruktor
    void przesun(double,double); // przesun
    void skaluj(double); // skaluj
    double odcieta(); // zwróć wartość x
    double rzedna(); // zwróć wartość y
};
```

Zadanie 3.

Napisz zewnętrzną funkcję Drukuj która będzie wyświetlała opis punktu i jego składowe. Jakie parametry należy przekazać tej funkcji?

Zadanie 4.

Uzupełnij klasę Punkt o możliwość przechowywania nazwy punktu. Załóż, że punkty opisujemy pojedynczymi literami alfabetu. Jakie zmiany będzie trzeba wykonać?

Zadanie 5.

Uzupełnij klasę Punkt o sprawdzanie czy dwa punkty mają takie same współrzędne.

Zadanie 6.

Napisz zewnętrzną funkcję, która sprawdza, czy dwa punkty znajdują się w takiej samej odległości od początku układu współrzędnych.

Zadanie 7.

Napisz program, w którym będziesz wykorzystywał tablicę punktów. Współrzędne punktów pobierane są z pliku. Czy klasa Punkt w obecnej postaci umożliwia budowę takiego programu? Co należałoby w niej zmienić?

Zadanie 8.

Na podstawie klasy Punkt utwórz klasę PunktB w taki sposób, aby w obiekcie zamiast współrzędnych kartezjańskich przechowywane były współrzędne biegunowe. Czy trzeba zmieniać interfejs publiczny tej klasy? Czy trzeba zmieniać program testujący klasę?

Zadanie 9.

Utwórz klasę Kwadrat, która będzie służyła do obsługi zbioru kwadratów na płaszczyźnie. Dane składowe obiektu to Lewy i Gorny - współrzędne lewego górnego wierzchołka, Bok - długość boku, Nazwa - nazwa kwadratu (jednoznakowa). Klasa ta ma zawierać:

- konstruktor domyślny Kwadrat::Kwadrat(), którego zadaniem jest przypisanie wartości domyślnych (0,1), bok 10 i nazwę pustą (spacja);
- konstruktor z jednym parametrem - nazwą Kwadrat::Kwadrat(char pNazwa), pozostałe parametry tak jak w konstruktorze domyślnym;
- konstruktor z pełnym zestawem parametrów Kwadrat::Kwadrat(int nLewy, int nGorny, int Bok, char pNazwa)
- funkcję składową void drukuj();
- funkcję składową void przesun(int dx, int dy).

Napisz program testowy, który utworzy trzy kwadraty (domyślny, domyślny z własną nazwą i zdefiniowany przez użytkownika), wyświetli ich charakterystyki, przesunie wybrany kwadrat i wyświetli jego położenie po przesunięciu.

Zadanie 10.

Posługując się klasą Kwadrat napisz program, za pomocą którego będzie można tworzyć zbiór kwadratów wyznaczony przez użytkownika. Użytkownik wpisuje: liczbę kwadratów (nie więcej niż liter w alfabecie), położenie i rozmiar boku poszczególnych kwadratów. Nazwą kwadratu jest kolejna litera alfabetu.

Program ma umożliwić wydrukowanie danych poszczególnych kwadratów, przesunięcie ich o 10 jednostek w prawo i do góry oraz ponowne wydrukowanie danych kwadratów.

Zadania domowe**Zadanie 11.**

Utwórz klasę Zbior_znakow, która będzie służyła do obsługi zbioru znaków ASCII. Nie używaj dynamicznego przydziału pamięci, zbiór zapamiętaj w tablicy o ustalonym z góry rozmiarze. Klasa ta ma zawierać:

- konstruktor, który zeruje tablicę ze zbiorem
- funkcję składową void zbior_dodaj(unsigned char c), która dodaje element do zbioru
- funkcję składową int zbior_nalezy(unsigned char c), która sprawdza, czy element należy do zbioru
- funkcję składową int zbior_licznosc(), która oblicza liczbę elementów zbioru.

Napisz program testowy, w którym za pomocą tej klasy będzie można sprawdzać liczbę różnych znaków zawartych we wprowadzonym słowie.

Zadanie 12.

Dany jest opis prostej listy:

- lista może przechowywać zero lub więcej elementów określonego typu
- można dodawać elementy do listy
- można sprawdzać, czy lista jest pusta
- można sprawdzać, czy lista jest pełna
- można odwiedzić każdy element listy i wykonać na nim jakąś operację.

Utwórz klasę odpowiadającą powyższemu opisowi. Zrealizuj listę jako tablicę i jako listę dowiązaniową. Publiczny interfejs klasy powinien być taki sam w obydwu wypadkach.