

# ZAAWANSOWANE PROGRAMOWANIE KOMPUTEROWE WNE (2023)

## 2. KLASY

(1) Zaimplementować klasę

```
class Punkt {
public:
    Punkt();
    Punkt(double _x, double _y);
    void drukuj();
    void przesun(double dx, double dy);
    double odleglosc(Punkt p);
    double x;
    double y;
};
```

Opis metod:

- `Punkt()`: tworzy punkt (0,0).
- `Punkt(double x, double y)`: tworzy punkt (x,y).
- `void drukuj()`: drukuje współrzędne na ekranie.
- `void przesun(double dx, double dy)`: przesuwa punkt o zadany wektor.
- `double odleglosc(Punkt p)`: zwraca odległość do podanego punktu.

(2) Zaimplementować klasę `Czas`. Obiekty tej klasy reprezentują czas w ciągu dnia z dokładnością do minuty.

```
class Czas {
public:
    Czas();
    Czas(int godzina, int minuta);
    void drukuj24();
    void drukuj12();
    int dodajMinuty(int ileMinut);
};
```

Opis metod:

- `Czas()`: tworzy północ (godzinę 0:00).
- `Czas(int godzina, int minuta)`: tworzy podaną godzinę.
- `void drukuj24()`: drukuje na ekranie w formacie 24-godzinnym.
- `void drukuj12()`: drukuje na ekranie w formacie 12-godzinnym.

- `int dodajMinuty(int ileMinut)`: dodaje podaną liczbę minut. Liczba minut może być ujemna.
- (3) Zaimplementować klasę `Macierz`, której obiekty reprezentują macierze o współczynnikach `double`. Zaimplementować operacje umożliwiające obliczenia na macierzach.
- (4) Zaimplementować klasę `Wymierna` umożliwiającą rachunki na liczbach wymiernych.
- (5) Zaimplementować klasę `Parking`. Obiekty tej klasy pozwalają zapamiętać, które miejsca na parkingu są wolne, a które zajęte. Miejsca są ponumerowane od 1 do  $n$ .

```
class Parking {
public:
    Parking(int ileMiejsc);
    void zajmij(int a);
    void zwolnij(int a);
    int ileWolnych();
    int znajdzWolne();
};
```

Opis metod:

- `Parking(int ileMiejsc)`: tworzy parking z miejscami  $1, \dots, \text{ileMiejsc}$ .
  - `void zajmij(int a)`: zajmuje miejsce o numerze  $a$ . Jeśli to miejsce jest już zajęte albo  $a$  nie jest poprawnym numerem miejsca, nic się nie dzieje.
  - `void zwolnij(int a)`: zwalnia miejsce o numerze  $a$ .
  - `int ileWolnych()`: zwraca liczbę wolnych miejsc.
  - `int znajdzWolne()`: zwraca numer dowolnego wolnego miejsca (lub 0 jeśli wszystkie są zajęte).
- (6) Zaimplementować klasę `Kolejka`, której obiekty reprezentują ciągi napisów typu `string`.

```
class Kolejka {
public:
    Kolejka();
    void dodaj(string a);
    string usun();
    int ile();
    string operator [] (int i);
};
```

Opis metod:

- `Kolejka()`: tworzy pustą kolejkę.
- `void dodaj(string a)`: dodaje napis na końcu kolejki.
- `string usun()`: usuwa i zwraca pierwszy napis.
- `int ile()`: zwraca liczbę elementów kolejki.
- `string operator [] (int i)`: zwraca napis o podanym indeksie.