

Zadanie 6

Zaprojektować i zaimplementować klasy *Punkt* i *Wielobok*.

Klasa *Punkt* ma odwzorowywać punkt na płaszczyźnie. Klasa powinna posiadać konstruktor domyślny oraz konstruktor z parametrami umożliwiający zainicjować pola składowe wartościami jego argumentów. Ponadto klasa powinna dostarczać metody dostępne do pól składowych umożliwiającymi zapis i odczyt wartości składowych. Klasa *Punkt* powinna posiadać metodę wyznaczającą odległość punktu od innego.

Klasa *Wielobok* powinna odwzorowywać wieloboczną figurę geometryczną na płaszczyźnie opisaną zbiorem punktów. Klasa powinna dostarczać konstruktor domyślny oraz konstruktor umożliwiający utworzenie zbioru punktów opisujących tę figurę. Ponadto klasa powinna dostarczać metody dostępne umożliwiające odczyt liczby punktów oraz dające możliwość odczytu i zapisu poszczególnych punktów. Klasa *Wielobok* powinna posiadać metodę zmiany zbioru punktów na nowy oraz metodę obliczania obwodu tej figury.

Przykład kodu testującego

```
Punkt p(2, 3);
std::cout << p.x() << ' ' << p.y() << '\n';
p.x() = 1;
p.y() = 1;
std::cout << p.x() << ' ' << p.y() << '\n';
std::cout << p.odleglosc(Punkt()) << '\n';
```

```
Punkt t[] = { Punkt(0, 1), Punkt(0, 0),
             Punkt(1, 0), Punkt(1, 1) };
Wielobok w(t, t+4);
std::cout << w.obwod() << '\n';;
```

```
w.punkt(1) = Punkt(0.5, 0.5);
std::cout << w.obwod() << '\n';;
```

```
Wielobok e;
e.punkty(t, t+3);
std::cout << e.obwod() << '\n';;
```

```
for (int i = 0; i < e.liczba(); ++i)
    std::cout << e.punkt(i).x() << ' '
              << e.punkt(i).y() << '\n';
```