

Cel ćwiczenia: **praktyczne zapoznanie się ze składnikami klasy będącymi obiektami,**
 praktyczne zapoznanie się ze wskaźnikami do elementów klasy

Należy napisać klasę `complex`, która będzie umożliwiała przechowywanie pojedynczej liczby zespolonej (liczba zespolona składa się z części rzeczywistej `re` oraz części zespolonej `im`). Klasa ma dysponować następującymi składnikami publicznymi:

1. konstruktorem domyślnym,
2. konstruktorami jedno i dwu argumentowym,
3. czterema bezargumentowymi funkcjami `fun1`, `fun2`, `fun3`, `fun4` typu `double` wykonującymi następujące działanie:
 - a. `fun1` – zwraca część rzeczywistą liczby zespolonej
 - b. `fun2` – zwraca część zespoloną liczby zespolonej
 - c. `fun3` – oblicza sumę kwadratu części rzeczywistej i urojonej ($re \cdot re + im \cdot im$)
 - d. `fun4` – oblicza pierwiastek z sumy kwadratu części rzeczywistej i urojonej ($\sqrt{re \cdot re + im \cdot im}$)
4. cztero elementową tablicą wskaźników (`tab_wsk_fun`) do bezargumentowych funkcji składowych klasy `complex` typu `double`. Tablica ta w konstruktorach ma być inicjalizowana tak by jej kolejne elementy zawierały wskaźniki do funkcji `fun1` – `fun4`.

Następnie należy utworzyć drugą klasę `tab_complex` zawierającą jako daną prywatną 10 elementową tablicę (`tab_c`) obiektów klasy `complex`. Klasa ma dysponować następującymi składnikami:

1. konstruktor domyślny – wypełniający elementy tablicy `tab_c` różnymi liczbami zespolonymi.
2. przeładowanym jako zaprzyjaźniona funkcja operatorem wyjścia `<<`. Działanie ma polegać na wypisaniu rezultatów wywołaniu (za pomocą tablicy wskaźników do funkcji `tab_wsk_fun` z klasy `complex`) dla każdego elementu tablicy `tab_c` funkcji `fun1` – `fun4` z klasy `complex`. Powyższe wywołanie ma zrealizowane za pomocą dwóch zagnieżdżonych pętli `for`. Zewnętrzna wykonuje się 10 razy i przechodzi przez kolejne elementy tablicy `tab_c`. Wewnętrzna wykonuje się 4 razy i dla każdego elementu tablicy `tab_c` wywołuje z pomocą tablicy `tab_wsk_fun` funkcje `fun1` – `fun4`.

Funkcja `main` ma utworzyć pojedynczy obiekt klasy `tab_complex` a następnie z wykorzystaniem operatora `<<` wypisać go na ekranie, poniżej kod funkcji `main`

```
int main(){
    tab_complex tc;
    cout << tc;
    return 0;
}
```

Oraz rezultat jej zadziałania dla przykładowych liczb zespolonych

```
0
0
0
0
1
1
2
1.41421
2
2
8
2.82843
3
3
18
4.24264
4
4
32
5.65685
5
5
50
7.07107
6
6
72
8.48528
7
7
98
9.89949
8
8
128
11.3137
9
9
162
12.7279
```