Zadanie 1

Utwórz klasę `Wektor2D` będącą implementacją wektorów w przestrzeni 2D. Program pisz drobnymi krokami. Utwórz najpierw pustą klasę bez funkcji i bez zmiennych składowych. Utwórz obiekt tej klasy w funkcji `main()`. Skompiluj i uruchom program. Dopiero teraz dodawaj kolejne elementy klasy kompilując program jak najczęściej.

- Klasa powinna zawierać 2 atrybuty w postaci zmiennych typu double opisujących współrzędne wektora. Zmienne te umieść w sekcji `private`.
- Utwórz konstruktory:
  - defaultowy: `Wektor2D()`
  - parametrowy: `Wektor2D(const double& xx, const double& yy)` (W jednym z konstruktorów użyj listy inicjalizacyjnej.)
- Utwórz metodę `Drukuj()` służącą do drukowania zawartości klasy na ekranie w formacie:
  wektor [2.345, 4.5678]
- W funkcji `main()` wykorzystaj klasę `Wektor2D` np.:

```
int main()
{
    Wektor2D v1(10,20);
    Wektor2D v2 = v1;
    Wektor2D v3(v1);
    Wektor2D v4;
    v1.Drukuj();
    v2.Drukuj();
    v3.Drukuj();
    v4.Drukuj();

    return 0;
}
```

- Wewnątrz konstruktorów drukuj informację o rodzaju właśnie wywoływanego konstruktora i wartościach składowych klasy.
- Przy pomocy debuggera prześledź wartość składowej `x` w poszczególnych funkcjach:
  - Postaw breakpoint na pierwszej linii kodu (*F9*).
  - Uruchom program w sesji debuggera (*F5*).
  - Po zatrzymaniu programu na "breakpoincie" wprowadź w oknie "watch" debuggera nazwę składowej klasy którą chcesz śledzić (np. `x`).
  - Wykonaj program etapami:
    - przechodząc do następnej linii kodu (*F10*);
    - lub wchodząc do środka funkcji (*F11*).
  - Odpowiedz sobie na następujące pytania:
    - Ile obiektów utworzyłeś w programie?
    - Ile konstruktorów zdefiniowałeś w klasie?
    - Do ilu konstruktorów udało Ci się wejść przy pomocy debuggera?
    - Które konstruktory kompilator utworzył automatycznie?
- Dodaj do klasy konstruktor kopiujący: `Wektor2D(const Wektor2D& v)` i powtórz doświadczenie z debuggerem.
- Dodaj do klasy destruktor. Wewnątrz destruktora umieść informacje identyfikujące obiekt który właśnie jest usuwany (np. wyświetlaj wartości składowych klasy).

Zadanie 2

Rozszerz klasę o obsługę wybranych operatorów.

- Utwórz operatory:
  - Jednoargumentowy `operator +=`
  - Dwuargumentowy `operator+` (sprawdź czy możesz operować na atrybutach klasy `Wektor2D` bez definiowania zaprzyjaźnienia z tą klasą)
- Dodaj i przetestuj w programie głównym następujące operacje:

```
v4 = v1 + v2;
v4.Drukuj();
v3 += v3;
v3.Drukuj();
```

Zadanie 3

Hermetyzacja i udostępnianie składowych klasy.

- Sprawdź czy możesz wykonać następującą operację w programie głównym:

```
v4.x = 50;
```

- Dodaj do klasy funkcję `SetX(const double& xx )` pozwalającą na zmianę składowej klasy `x`. Ogranicz wartość tej zmiennej do `MAX_VAL = 100`. Przekroczenie wartości `MAX_VAL` powinno kończyć program.
- Dodaj do klasy funkcje `GetX()` zwracającą wartość składowej `x`.
- Dodaj analogiczne funkcje dla składowej `y` i wykonaj następujący fragment kodu w programie głównym

```
v4.SetX(50);
v4.Drukuj();
Wektor2D v5( v4.GetY(), v3.GetX() );
v5.Drukuj();
```

Zadanie 4

Zmienne statyczne klasy. * Dodaj do klasy prywatną składową statyczną `indx`. Zmienna ta powinna nadawać kolejne numery tworzonym obiektom. Zmienna taka musi być zainicjalizowana poza klasą w następujący sposób:

```
int Wekro2D::indx = 0;
```

- Dodaj do klasy prywatną składową `nr`. Składowa ta ma przechowywać kolejny numer obiektu (począwszy od 1). Inicjalizuj ją przy pomocy składowej `indx` w każdym dostępnym konstruktorze jednocześnie zwiększając wartość składowej `indx`.
- Wartość zmiennej `nr` wyświetlaj zarówno w konstruktorach jak i w destruktorze.
- Uruchom program i przeanalizuj jeszcze raz kolejność tworzenia się i usuwania obiektów.
  - Zwróć szczególną uwagę na moment konstrukcji i destrukcji obiektów o $nr = 5$ i $nr = 6$.
  - Jak zmieni się liczba tworzonych obiektów jeśli konstruktor kopiujący zmienisz na następujący: `Wektor2D(const Wektor2D v)` (czyli bez referencji).
  - Jak wytłumaczysz, że przy destrukcji obiektów nigdzie nie pojawia się wartość $nr = 4$?
- Dodaj do klasy odpowiednio zdefiniowany operator przypisania(=). Czy teraz, w trakcie usuwania obiektów, widać już wartość $nr = 4$?

Zadanie 5

Zmodyfikuj program tak aby każda klasa była umieszczona w oddzielnym pliku .h i .cpp.