

Programowanie obiektowe dla AiR - Kolokwium 1 - 15 kwietnia 2024 r.

Rozwiązanie kolokwium, w postaci plików źródłowych *.cpp i nagłówkowych *.h, umieść w Moodle <https://moodle.umk.pl/mod/assign/view.php?id=207481>

Zaliczenie wymaga wykonanie co najmniej 50% programu zgodnie z poniższym opisem.

Zadanie: Linia łamana.

Napisz program wyznaczający długość linii łamanej składającej się z sekwencji punktów na płaszczyźnie. Program implementuje dwie klasy, **Punkt** i **Linia**, zgodnie z poniższą specyfikacją:

Klasa **Punkt** reprezentuje punkt na płaszczyźnie.

Punkt zawiera następujące atrybuty:

- **x, y**: współrzędne punktu określone liczbami rzeczywistymi

Punkt udostępnia następujące operacje:

- konstruktor domniemany tworzący punkt o współrzędnych (0,0)
- konstruktor inicjujący punkt o współrzędnych (*x, y*) podanych w argumentach
- funkcję składową **Dystans(const Punkt &p)**, która zwraca odległość (liczba rzeczywista) do wskazanego w argumencie punktu **p**. Odległość między punktami (*x₁, y₁*) i (*x₂, y₂*) dana jest wzorem

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

- funkcję składową **Dystans()**, która zwraca odległość od środka układu współrzędnych (0,0)
- funkcję operator<<, która przeciążając operator przesunięcia bitowego << w taki sposób aby możliwe było wypisanie punktu na strumieniu wyjściowym **ostream** w postaci napisu "Punkt (1.3 , -2.4)"

Klasa o nazwie **Linia** reprezentuje linię łamaną utworzoną przez sekwencję punktów na płaszczyźnie.

Linia posiada następujące atrybuty:

- **ile**: ilość punktów, liczba całkowita
- **punkty**: adres tablicy zawierającej sekwencję punktów (obiektów typu **Punkt**)

Linia łamana udostępnia następujące operacje:

- konstruktor z jednym argumentem typu **int** określającym ilość punktów tworzących linię łamaną. Konstruktor alokuje tablicę punktów o podanym rozmiarze, wszystkie punkty znajdują się w środku układu współrzędnych (0,0)
- konstruktor kopiujący
- funkcję składową **Wczytaj()**, która wczytuje z konsoli (ze strumienia **std::cin**) współrzędne wszystkich punktów tworzących linię łamaną.
- funkcję składową **Dlugosc()**, która zwraca liczbę rzeczywistą równą długości linii łamanej, czyli sumę odległości pomiędzy kolejnymi punktami tworzącymi linię
- przeciążony operator przesunięcia bitowego << w taki sposób aby możliwe było wypisanie linii na strumieniu wyjściowym **ostream** w postaci ponumerowanej listy zawierającej punkty składowe, np.:

Linia

Ilosc punktow 3

1: Punkt (1.1, 3.14)

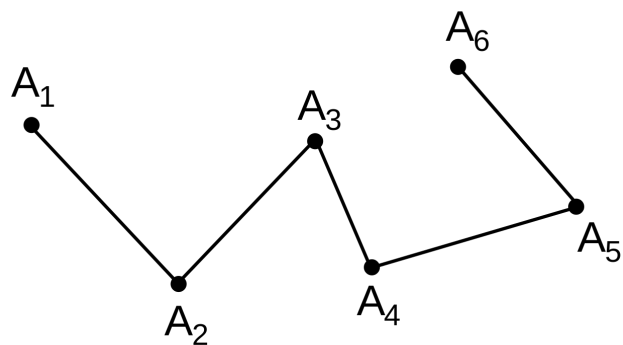
2: Punkt (-2.5, 0.1)

3: Punkt (4, 0)

Wykorzystaj klasy **Punkt** oraz **Linia** do realizacji programu, który wykona następujące operacje:

1. użytkownik podaje ilość punktów tworzących linię łamaną
2. program utworzy linię łamaną o liczbie punktów podanej przez użytkownika
3. użytkownik wprowadza współrzędne wszystkich punktów
4. program wypisuje na wyjściu informacje o długości linii łamanej oraz listę współrzędnych punktów tworzących linię

Pamiętaj o hermetyzacji klas (pola składowe powinny być prywatne), podziale projektu na pliki nagłówkowe i źródłowe, jeśli uznasz za potrzebne to zaimplementuj dodatkowe mechanizmy, np. destruktory, przeciążania operatorów, deklaracje przyjaźni, itp.



Przykład działania programu:

Ile punktów ? 3

Wsp (x, y) punktu 1 : 1.1 3.14

Wsp (x, y) punktu 2 : -2.5 0.1

Wsp (x, y) punktu 3 : 4 0

Długość linii 11.2126

Linia

Ilość punktów 3

1: Punkt (1.1, 3.14)

2: Punkt (-2.5, 0.1)

3: Punkt (4, 0)