

ZADANIE 4

Napisać program pracujący na parabolach.

Funkcjonalność:

Użytkownik wprowadza z klawiatury współrzędne 3 punktów, następnie w oparciu o wprowadzone dane sprawdzane jest, czy można przez nie przeprowadzić parabolę. W przypadku, kiedy jest to możliwe, użytkownik podaje wartość współrzędnej odciętej (x), aby przeprowadzić styczną do paraboli.

Funkcja główna ma wyglądać następująco:

- 1.wprowadzić współrzędne 3 punktów, wypisać informację o wprowadzonych punktach
- 2.sprawdzić, czy jest możliwe przeprowadzenie przez punkty paraboli (jeśli nie, program ma zakończyć swoje działanie)
- 3.wypisać informację (równanie) paraboli postaci $y = ax^2 + bx + c$
- 4.podać wartość współrzędnej odciętej punktu, gdzie ma zostać poprowadzona styczna do paraboli, wyliczyć wartość rzędnej punktu styczności, wypisać informację o punkcie styczności
- 5.wyliczyć równanie stycznej do paraboli postaci $y = ax + b$

Oprogramowanie:

Należy stworzyć klasę: **point**, **parabola**

Klasa **point** ma zawierać następujące pola prywatne: x , y , pole statyczne: licznik *counter* obiektów klasy.

Klasa **point** ma zawierać następujące pola publiczne: konstruktor (główny i domniemany), destruktory, funkcję wypisującą informację o produkcie, funkcję pozwalającą wyliczyć wartość współrzędnej y z równania krzywej przekazanego w argumencie funkcji, funkcję statyczną wypisującą licznik punktów.

Klasa **parabola** ma zawierać następujące pola prywatne: a , b , c , pole statyczne: licznik *counter* obiektów klasy.

Klasa **parabola** ma zawierać następujące pola publiczne: konstruktor (główny i domniemany), destruktory, funkcję wypisującą informację o produkcie, funkcję pozwalającą współczynniki paraboli w oparciu o współrzędne 3 punktów podanych w argumencie funkcji, funkcję statyczną wypisującą licznik parabol.

Należy pamiętać o zaprzyjaźnieniu odpowiednich klas ze sobą w celu umożliwienia dostępu do pól prywatnych.

W funkcji głównej tworzone są wskaźniki (nie obiekty!) punktów oraz parabol z odpowiednimi rezerwacjami pamięci. Należy pamiętać, aby w chwili, gdy program ma zakończyć swoje działanie została zwolniona przydzielona pamięć. Zwolnienia pamięci dokonujemy w odwrotnej kolejności, jak była przydzielana.

Nie ma potrzeby (choć można!) deklarowania klasy **line**.

Należy natomiast napisać globalną, zewnętrzną funkcję zaprzyjaźnioną z odpowiednimi klasami pozwalającą policzyć wartości współczynników stycznej do paraboli (niech wyliczone współczynniki stycznej będą zwracane w argumentach funkcji).

Przypomnienie:

Równanie paraboli: $f(x) = y = ax^2 + bx + c$

Mając 3 punkty $A = (x_A, y_A)$; $B = (x_B, y_B)$; $C = (x_C, y_C)$

mamy do rozwiązania układ 3 równań z 3 niewiadomymi: a, b, c (należy rozwiązać na kartce papieru)

Punkt styczności: $D = (x_D, y_D)$ – współrzędną y_D liczymy z równania paraboli

Styczna: $g(x) = y = cx + d$

współczynnik kierunkowy stycznej c jest równy wartości pierwszej pochodnej równania paraboli w punkcie styczności D :

$c = f'(x = x_D)$

wyraz wolny stycznej d liczymy pamiętając, że styczna przechodzi przez punkt styczności D .

Program może być kontynuowany na kolejnych zajęciach!

Program należy napisać w trzech plikach: nagłówkowym (*.h), implementacyjnym (*.cpp), oraz z funkcją główną (*main.cpp).