Список 01 – Домашнее Задание

Классы и объекты

Предмет: Алгоритмизация и программирование

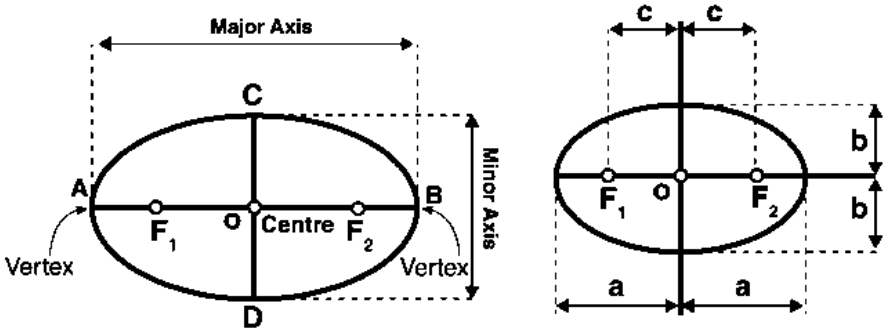
Преподаватель: Хольгер Эспинола Ривера

1. Класс эллипса. Принимая во внимание графики выше, сформулируйте класс на C++, который содержит атрибуты и методы, характеризующие эллипс.

Эллипс в аналитической геометрии обычно определяется координатами его центра C(h, k), длинами большой полуоси **a** и малой полуоси **b**. В зависимости от того, где расположена большая ось, уравнения эллипса могут быть определены следующими уравнениями:

a) если большая ось принадлежит оси **X**: $\frac{(x-h)^2}{a^2}+\frac{(y-k)^2}{b^2}=1$

b) если большая ось принадлежит оси **Y**: $\frac{(x-h)^2}{b^2}+\frac{(y-k)^2}{a^2}=1$



Реализовать следующие реализации:

[1]. Составить UML-диаграмму, определив атрибуты и методы для класса Ellipsis.

[2]. Разработать конструкторы по параметрам, по умолчанию и по копии объекта. В конструкторе по умолчанию, если центр эллипса не определен, предположить, что координаты центра — C (0, 0). Также в конструкторе необходимо определить ограничения, гарантирующие, что значение большой полуоси **a** всегда должно быть больше значения малой полуоси **b**, в противном случае потребуется запрос на замену значений для **a** и **b**.

[3]. Разработайте функцию для вычисления гиперпараметра эллипсиса **c**, используя уравнение: $a^2=b^2+c^2$

[4]. Реализовать методы запроса координат основных характерных точек эллипса:

- Вершины: $V_1(h-a,k)$ и $V_2(h+a,k)$ для большой оси в X или $V_1(h,k-a)$ и $V_2(h,k+a)$ для большой оси Y

- Фокус: $F_1(h-c,k)$ и $F_2(h+c,k)$ для большой оси в X или $F_1(h,k-c)$ и $F_2(h,k+c)$ для большой оси Y

[5]. Реализовать методы запроса значений характеристических параметров многоточия:

- Длина фокальной хорды: $LR=\frac{2b^2}{a}$ - эксцентриситет: $e=\frac{c}{a}$

[6]. Реализуйте методы **set** и **get** для класса Ellipse. Используйте метод set для передачи значений аргументов для изменения значений атрибутов в классе и используйте метод get для извлечения значений частных переменных класса.

[7]. Реализуйте процедуру для печати общего уравнения эллипса и их характерных точек: центра C, вершин V и фокусов F.

[8]. Реализуйте функцию для проверки того, находится ли некоторая точка P (x, y) внутри, над или снаружи эллипса.

[9]. Реализуйте функцию для вычисления приблизительного периметра эллипса: $P=\pi\left[3(a+b)-\sqrt{(3a+b)(a+3b)}\right]$

[10]. Реализуйте функцию для вычисления площади эллипса: $S=\pi\cdot ab$

[11]. Реализовать функцию для заданной некоторой координаты общей точки E, которая принадлежит эллипсу, способную вычислить вторую координату, используя общие уравнения эллипса. Если функция, координата **x** дана, необходимо вычислить вторую координату **y**, и тоже наоборот.

2. Класс кватернион. Кватернионы — это математическая числовая система, которая является расширением комплексных чисел, образуя гиперкомплексную числовую систему, которая работает в 4-мерном пространстве. Кватернионное число определяется формой:

$$q=a+b\vec{i}+c\vec{j}+d\vec{k}\text{, где:}$$

a: действительная часть (скаляр)

b, c, d: действительные коэффициенты, связанные с каждой мнимой единицей

i, j, k: различные мнимые единицы, составляющие векторную часть кватерниона, удовлетворяющую условию $i^2 = j^2 = k^2 = ijk = -1$

Учитывая свойства числового класса кватернионов, реализуем класс Quaternion на языке C++, реализующий следующие реализации:

[1]. Составляем диаграмму UML, определяющую атрибуты и методы для класса Quaternion.

[2]. Разрабатываем конструкторы по параметрам, по умолчанию и по копированию объектов. В конструкторе по умолчанию необходимо инициализировать вещественное число 1, а комплексные компоненты — нулем.

[3]. Реализуем методы-сеттеры для передачи значений аргументов объекту класса и геттеры для извлечения аргументов кватерниона.

[4]. Реализуем процедуру для печати кватерниона в типичном числовом формате.

[5]. Реализуем функцию для вычисления нормы кватерниона, заданной формулой: $\|q\| = \sqrt{a^2 + b^2 + c^2 + d^2}$

[6]. Реализовать функцию для получения сопряжения кватерниона. Учитывая стандартный формат кватерниона, сопряжение задается как:

$$\bar{q} = a - b\vec{i} - c\vec{j} - d\vec{k}$$

[7]. Реализовать функцию для получения операции нормализации кватерниона, заданного формулой: $norm(q) = \frac{q}{\|q\|}$

[8]. Реализуйте функцию для вычисления инверсии кватерниона.

Формулировка имеет вид: $q^{-1} = \frac{\bar{q}}{\|q\|^2}$

[9]. Реализовать в качестве глобальных функций в C++ операции между двумя кватернионными числами (нужно исследовать, как выполнять эти операции в этой области чисел):

- добавление: $q_1 + q_2$

- вычитание: $q_1 - q_2$

- умножение: $q_1 \times q_2$

- разделение: $q_1 \times (q_2)^{-1}$

- скалярное произведение: $q_1 \cdot q_2$

[10]. Реализовать функцию для вычисления евклидова расстояния между двумя кватернионными числами, заданными по формуле:

$$d(q_1, q_2) = \sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2 + (c_1 - c_2)^2 + (d_1 - d_2)^2}$$

[11]. Реализовать функцию для вычисления нормы (нормы Чебышева) между двумя кватернионными числами, заданными формулой:

$$L_\infty(q_1, q_2) = \max\left\{|a_1 - a_2|, |b_1 - b_2|, |c_1 - c_2|, |d_1 - d_2|\right\}$$