#include <iostream>

#include <cmath>

class Point { // Объявляем класс Point для представления точки в декартовой системе координат

private:

double x, y; // Приватные поля для хранения координат x и y

public:

Point(double x, double y) : x(x), y(y) {} // Конструктор класса Point, который инициализирует координаты x и y

double getX() const { // Публичный метод для получения координаты x

return x;

}

double getY() const { // Публичный метод для получения координаты y

return y;

}

static Point fromPolar(double r, double theta) { // Статический метод для создания точки по полярным координатам

double x = r \* cos(theta); // Вычисляем координату x по полярным координатам

double y = r \* sin(theta); // Вычисляем координату y по полярным координатам

return Point(x, y); // Возвращаем созданную точку

}

};

class Hexagon { // Объявляем класс Hexagon для представления правильного шестиугольника

private:

double side; // Приватное поле для хранения длины стороны

public:

Hexagon(double side) : side(side) {} // Конструктор класса Hexagon, который инициализирует длину стороны

void printVertices() { // Публичный метод для вывода координат вершин шестиугольника

double angle = 0; // Устанавливаем начальный угол

for (int i = 0; i < 6; ++i) { // Цикл для вычисления координат вершин шестиугольника

Point vertex = Point::fromPolar(side, angle); // Создаем точку, используя метод fromPolar класса Point

std::cout << "Vertex " << i + 1 << " : (" << vertex.getX() << ", " << vertex.getY() << ")" << std::endl; // Выводим координаты вершины

angle += M\_PI / 3; // Увеличиваем угол на 60 градусов в радианах

}

}

};

int main(int argc, char \*argv[]) { // Основная функция программы

if (argc < 2) { // Проверяем, передан ли аргумент командной строки

std::cerr << "Usage: " << argv[0] << " <side length>" << std::endl; // Выводим сообщение об использовании программы

return 1; // Завершаем программу с ошибкой

}

double sideLength = std::stod(argv[1]); // Получаем длину стороны из аргумента командной строки

Hexagon hexagon(sideLength); // Создаем объект Hexagon с заданной длиной стороны

hexagon.printVertices(); // Выводим координаты вершин шестиугольника

return 0; // Завершаем программу успешно

}

Int main