

## 1. Код

```
/*
 * Для некоторой простоты я предполагал, что конус круговой, расположен вертикально и
 * вершина выше чем основание. Плюс радиус основания и высота предполагаются положительными,
 * но никакой проверки не делается.
 */

#include <iostream>

using namespace std;

class Point {
private:
    double x;
    double y;
    double z;
public:
    Point();
    Point(double, double, double);
    ~Point();
    double getX() { return x; }
    double getY() { return y; }
    double getZ() { return z; }
    void setX(double x_) { x = x_; }
    void setY(double y_) { y = y_; }
    void setZ(double z_) { z = z_; }
    void printPoint();
    void inputPoint();
};

class Cone {
private:
    Point center;
    double radius;
    double height;
public:
    Cone();
    Cone(double, double, double, double, double);
    ~Cone();
    double getRadius() { return radius; }
    double getHeight() { return height; }
    double getCenterX() { return center.getX(); }
    double getCenterY() { return center.getY(); }
    double getCenterZ() { return center.getZ(); }
    void printCone();
    void inputCone();
    bool isIn(Point);
};

bool isIn(Point p, Cone c) {
    if ((p.getZ() < c.getCenterZ()) || (p.getZ() > c.getCenterZ() + c.getHeight())) {
        return false;
    }
    if ((c.getRadius() * (p.getZ() - c.getCenterZ()) + c.getHeight() * sqrt(pow((p.getX()
- c.getCenterX()), 2) + pow((p.getY() - c.getCenterY()), 2)) - c.getRadius() *
c.getHeight()) > 0) {
```

```

        return false;
    }
    return true;
}

int main()
{
    Point p;
    Cone c;
    p.inputPoint();
    c.inputCone();
    // cout << c.isIn(p) << endl;
    string result = (isIn(p, c)) ? "Yes" : "No";
    cout << endl << "Q: Is the point inside of the cone?\nA: " << result << endl;
    p.printPoint();
    c.printCone();
    return 0;
}

Point::Point() { // default constructor
    x = 0.0;
    y = 0.0;
    z = 0.0;
}

Point::Point(double a, double b, double c) { // constructor with parameters
    x = a;
    y = b;
    z = c;
}

Point::~~Point() { // destructor. It's empty because we didn't allocate any memory with
new
}

void Point::printPoint() {
    cout << endl << "Point coordinates is (" << x << ", " << y << ", " << z << ")" <<
endl;
}

void Point::inputPoint() {
    cout << "Enter point coordinates" << endl;
    cin >> x >> y >> z;
}

Cone::Cone() { // default constructor
    Point center(0, 0, 0);
    radius = 1.0;
    height = 1.0;
}

Cone::Cone(double x_, double y_, double z_, double radius_, double height_) {
    center.setX(x_);
    center.setY(y_);
    center.setZ(z_);
    radius = radius_;
    height = height_;
}

```

```

}

Cone::~Cone() { // destructor. Same as Point class, didn't allocate any memory, so there
is nothing to delete

}

void Cone::printCone() {
    cout << endl << "    Cone" << endl;
    cout << "Center of the base: (" << getCenterX() << ", " << getCenterY() << ", " <<
getCenterZ() << ")" << endl;
    cout << "Radius of the base: " << radius << endl;
    cout << "Height:          " << height << endl;
}

void Cone::inputCone() {
    double x_, y_, z_;
    cout << "Enter cone parameters (coordinates of center of base, radius, height):" <<
endl;
    cin >> x_ >> y_ >> z_ >> radius >> height;
    center.setX(x_);
    center.setY(y_);
    center.setZ(z_);
}

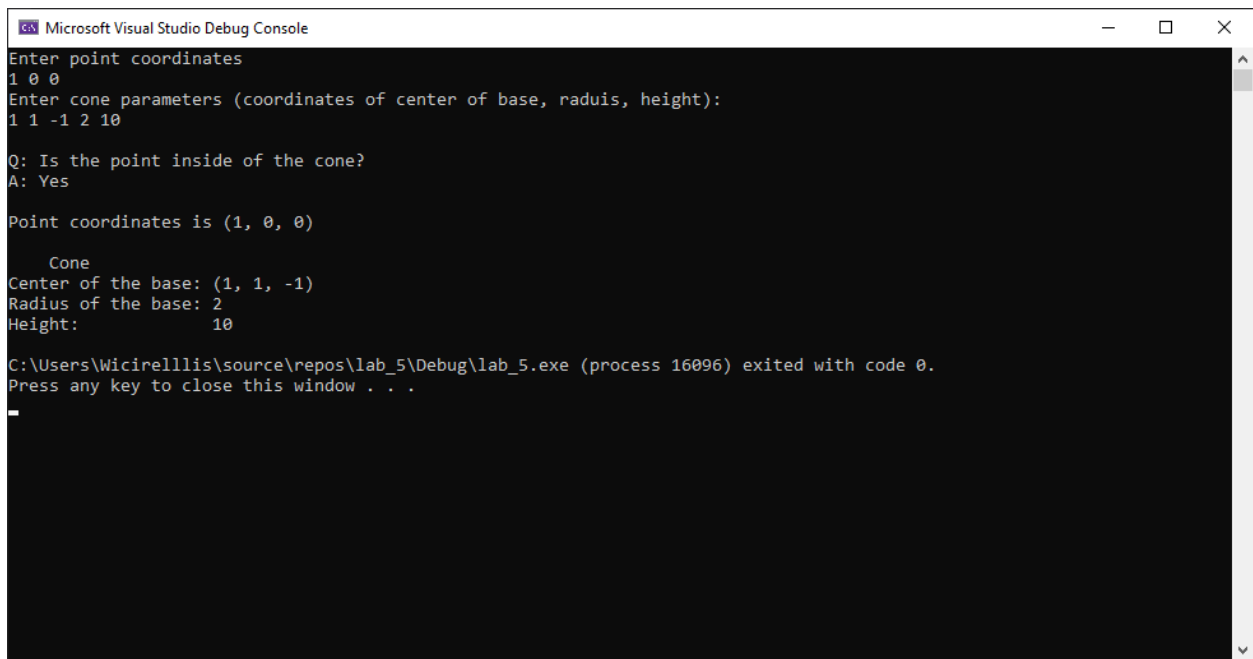
bool Cone::isIn(Point p) { // some geometry-based checks, it should work
    if ((p.getZ() < center.getZ()) || (p.getZ() > center.getZ() + height)) {
        return false;
    }
    if ((radius * (p.getZ() - center.getZ()) + height * sqrt(pow((p.getX() -
center.getX()), 2) + pow((p.getY() - center.getY()), 2)) - radius * height) > 0) {
        return false;
    }
    return true;
}

```

2.

Проверим работу программы на каком-нибудь входе. Пусть ввод следующий:  
1 0 0 (точка)

1 1 -1 2 10 (конус, первые три числа – это координаты центра основания,  
потом радиус основания, потом высота)

A screenshot of the Microsoft Visual Studio Debug Console window. The window has a title bar with the text "Microsoft Visual Studio Debug Console" and standard window controls (minimize, maximize, close). The console output is as follows:

```
Enter point coordinates
1 0 0
Enter cone parameters (coordinates of center of base, radius, height):
1 1 -1 2 10

Q: Is the point inside of the cone?
A: Yes

Point coordinates is (1, 0, 0)

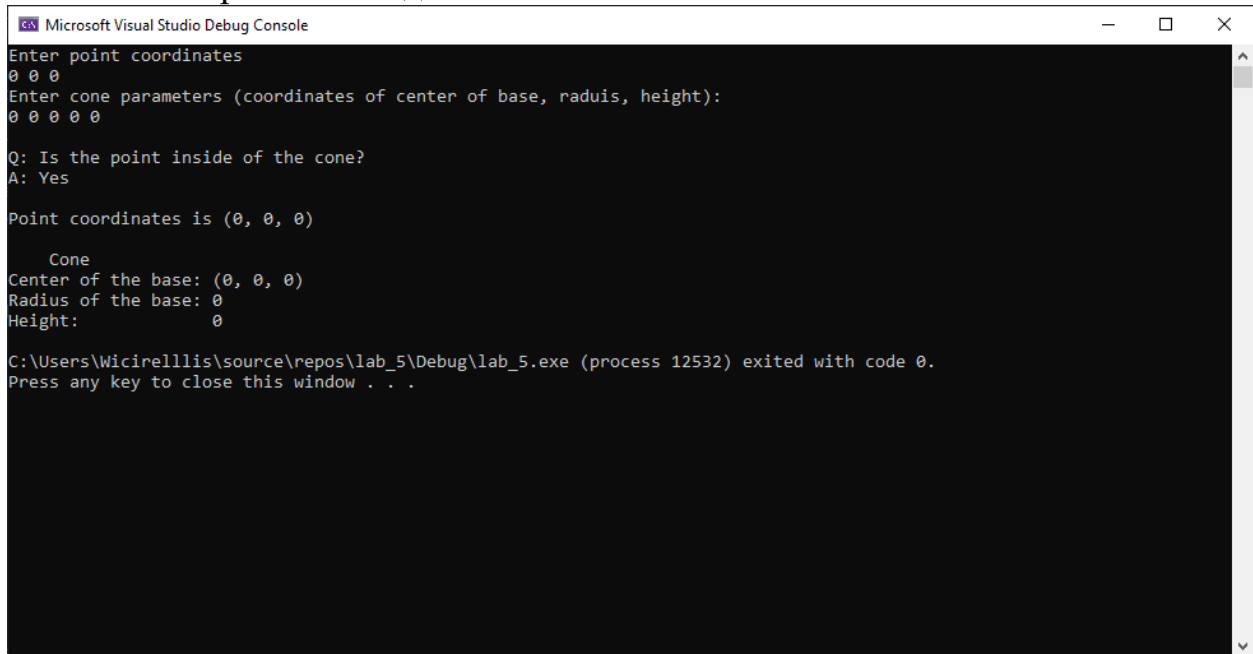
      Cone
Center of the base: (1, 1, -1)
Radius of the base: 2
Height:           10

C:\Users\wicirellis\source\repos\lab_5\Debug\lab_5.exe (process 16096) exited with code 0.
Press any key to close this window . . .
```

Работает (по крайней мере на данном входе) верно.

3.

## Несколько скринов вывода



```
Microsoft Visual Studio Debug Console

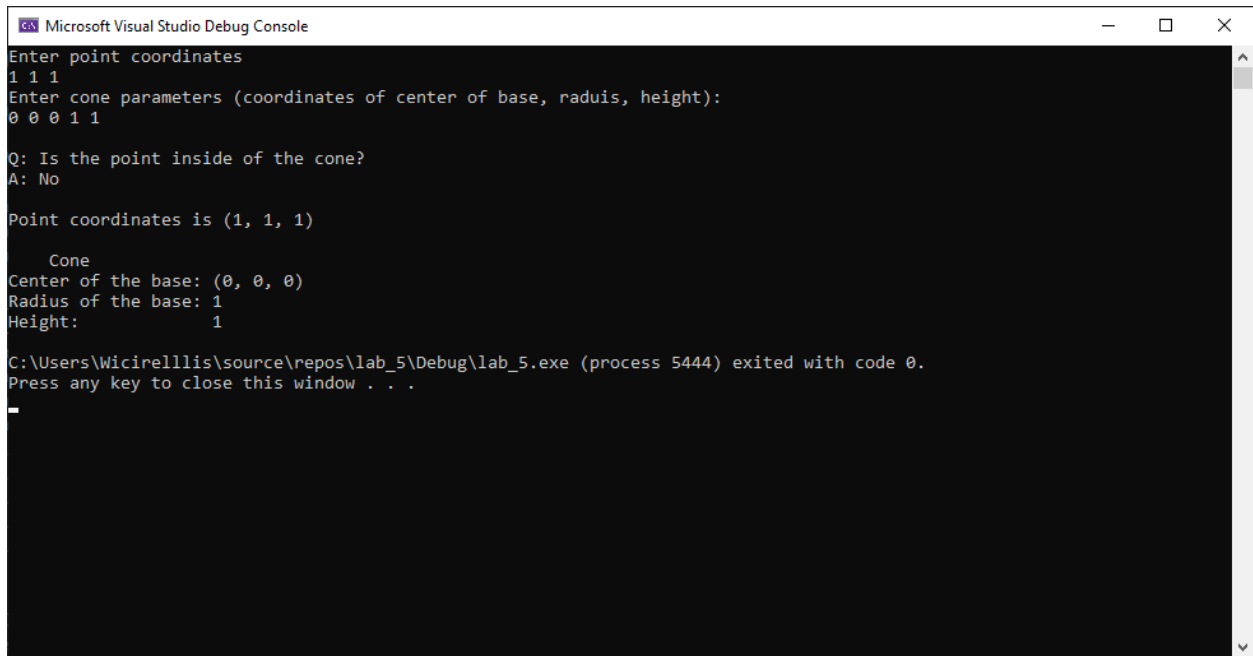
Enter point coordinates
0 0 0
Enter cone parameters (coordinates of center of base, radius, height):
0 0 0 0 0

Q: Is the point inside of the cone?
A: Yes

Point coordinates is (0, 0, 0)

      Cone
Center of the base: (0, 0, 0)
Radius of the base: 0
Height:           0

C:\Users\Wicirelllis\source\repos\lab_5\Debug\lab_5.exe (process 12532) exited with code 0.
Press any key to close this window . . .
```



```
Microsoft Visual Studio Debug Console

Enter point coordinates
1 1 1
Enter cone parameters (coordinates of center of base, radius, height):
0 0 0 1 1

Q: Is the point inside of the cone?
A: No

Point coordinates is (1, 1, 1)

      Cone
Center of the base: (0, 0, 0)
Radius of the base: 1
Height:           1

C:\Users\Wicirelllis\source\repos\lab_5\Debug\lab_5.exe (process 5444) exited with code 0.
Press any key to close this window . . .
```

## 4. Вопросы

### 1) Что такое класс? Что класс может в себе хранить?

Класс – это фактически структура с методами (=функциями). Т.е. в классе есть данные (переменные базовых типов, структуры, другие классы), и функции. Хранить класс может все что угодно.

### 2) Какие спецификаторы доступа существуют и в чем их особенности?

Private, public, protected.

Private – используется по-умолчанию, доступ есть только у членов класса.

Например прайват переменную можно прочитать из любого метода класса, но снаружи нельзя. Прайват определяется для класса, а не экземпляра класса.

Public – доступно всем.

Protected – подобно прайват, доступ есть из самого класса и наследованных от него.

### 3) Что такое метод? Для чего нужны конструктор и деструктор? Как они задаются?

Метод – это просто функция привязанная к классу.

Конструктор – это метод, который вызывается при создании экземпляра класса. Называется также как и класс. Может иметь параметры. Можно перегрузить конструктор, задав несколько вариантов, в зависимости от количества и типа параметров.

Деструктор вызывается при удалении экземпляра класса. Нужен, например, чтобы освободить за собой память. Называется как класс с тильдой (~) в начале. Деструктор может быть только один.

Конструктор и деструктор не возвращают значение, поэтому их объявление выглядит как-то так

```
class MyClass {  
public:  
    int add()  
    MyClass();  
    MyClass(int, int);  
    ~MyClass();  
};
```

В этом примере add – это просто метод класса, после него идет конструктор без параметров, потом конструктор с двумя параметрами, потом деструктор.