

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего  
профессионального образования  
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО  
ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
РУКОВОДИТЕЛЬ

Старший преподаватель  
должность, уч. степень, звание

\_\_\_\_\_  
подпись, дата

Е.О.Шумова  
инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

№2

**«Классы, конструкторы, деструкторы»**

по дисциплине: Объектно-ориентированное программирование

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. Z7431

20.01.2020  
подпись, дата

М.Д.Семочкин  
инициалы, фамилия

Студ. билет 2014/1054

Санкт-Петербург  
2020

## **Цель работы**

Изучить принципы создания классов с конструкторами, применив на практике знания базовых синтаксических конструкций языка C++ и объектно-ориентированного программирования.

Закрепить знания по теме: Классы, конструкторы, деструкторы

## **Задание**

В работе необходимо реализовать класс в соответствии с вариантом задания и создать приложение. В классе должен быть предусмотрен конструктор для установки начальных значений полей, конструктор по умолчанию, конструктор копирования, деструктор. Поля класса должны иметь спецификатор доступа `private`. Поля могут быть объявлены, как указатели. При использовании указателей в конструкторе класса должна быть инициализация полей, через оператор `new`, в деструкторе освобождение памяти, через оператор `delete`. Доступ к полям осуществляется через `public` методы. В функции `main()` создается не менее 3 объектов класса (с использованием всех конструкторов) и осуществляется вызов методов класса.

Вариант 17:

Определите класс `CYL`, который имеет конструктор с двумя параметрами (высота и радиус). Создать функцию-член, вычисляющую объем цилиндра. Если в конструктор не передаются параметры, то вычислить объем куба со стороной 5.

## **Используемые инструменты разработки**

IDE – Microsoft Visual C++ 2010 Express

## Текст программы

```
// В работе необходимо реализовать класс. В классе должен быть предусмотрен
// конструктор для установки начальных значений полей, конструктор по умолчанию,
// конструктор копирования, деструктор.
//
// Поля класса должны иметь спецификатор доступа private.
// Доступ к полям осуществляется через public методы.
//
// Поля могут быть объявлены, как указатели.
// При использовании указателей в конструкторе класса должна быть инициализация полей,
// через оператор new, в деструкторе освобождение памяти, через оператор delete.
//
// В функции main() создается не менее 3 объектов класса (с использованием всех
конструкторов)
// и осуществляется вызов методов класса.
//
//
// Вариант 17
//
// Определите класс CYL, который имеет конструктор с двумя параметрами (высота и радиус).
// Создать функцию-член, вычисляющую объем цилиндра.
// Если в конструктор не передаются параметры, то вычислить объем со стороной 5.
```

```
#include <iostream>
#include <stdlib.h>
using namespace std;
```

```
class CYL {
    private:
        float height;
        float radius;

    public:
        // Конструктор по умолчанию
        CYL();
        // Конструктор с параметрами
        CYL(float h, float r);
        // Конструктор копирования
        CYL(const CYL& other);
        // Деструктор
        ~CYL();
        void print();
        float getVolume();
};
```

```
CYL::CYL() {
    height = 5;
    radius = 5;
}
```

```
CYL::CYL(float h, float r) {
    height = h;
    radius = r;
}
```

```

CYL::CYL(const CYL& other) {
    this->height = other.height;
    this->radius = other.radius;
}

CYL::~~CYL() {
    // Среди атрибутов класса только примитивы, поэтому ничего не
    // нужно удалять в деструкторе вручную
    cout << "Объект удален" << endl;
}

void CYL::print() {
    cout << "height: " << height << endl << "radius: " << radius << endl << endl;
}

float CYL::getVolume() {
    return 3.14 * radius * radius * height;
}

int main(int argc, char **argv) {
    setlocale(LC_ALL, "russian");

    // Создаем объект с помощью конструктора по умолчанию
    CYL cylinder1;
    cylinder1.print();
    // Создаем копию объекта (при этом используется созданный ранее конструктор
    копирования)
    CYL cylinder1_clone = cylinder1;
    cylinder1_clone.print();

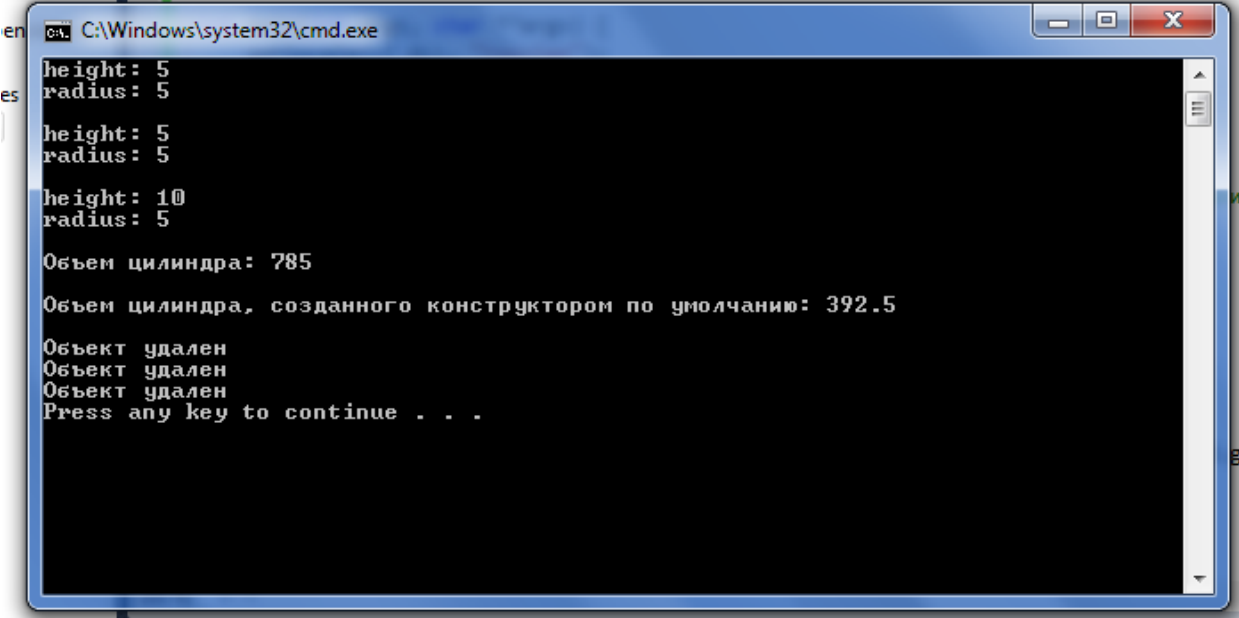
    // Создаем объект с помощью конструктора с параметрами
    CYL cylinder2(10, 5);
    cylinder2.print();
    cout << "Объем цилиндра: " << cylinder2.getVolume() << endl << endl;

    cout << "Объем цилиндра, созданного конструктором по умолчанию: " <<
    cylinder1.getVolume() << endl << endl;

    // Перед выходом из блока main для объектов будет вызван их деструктор
}

```

## Результат работы программы



```
C:\Windows\system32\cmd.exe
height: 5
radius: 5
height: 5
radius: 5
height: 10
radius: 5
Объем цилиндра: 785
Объем цилиндра, созданного конструктором по умолчанию: 392.5
Объект удален
Объект удален
Объект удален
Press any key to continue . . .
```

Output

## Вывод

Во время выполнения лабораторной работы были получены навыки использования конструкторов и деструкторов при создании классов на языке C++.