ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | Е.О. Шумова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| «Классы, определение методов класса, права доступа» |
| по курсу: ОБЪЕКТНО ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4134к |  |  |  | Костяков Н.А. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

**Цель работы**

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

**Вариант** 3 Определите класс cyl для хранения информации о цилиндре. Класс должен содержать конструктор задающий радиус и высоту цилиндра и функции вычисляющие объем и полную площадь цилиндра.

**Ход выполнения**

**Листинг**

#include <iostream>

#define \_USE\_MATH\_DEFINES

#include <math.h>

class cyl

{

public:

cyl(double radius = 0 , double height = 0);

cyl(const cyl& obj);

~cyl();

double volume();

double square();

private:

double radius, height;

};

cyl::cyl(double radius, double height)

{

std::cout << "Вызов контсруктора " << this<<std::endl;

this->radius = radius;

this->height = height;

}

cyl::cyl(const cyl& obj)

{

std::cout << "Вызов контсруктора копирования " << this << std::endl;

this->radius = obj.radius;

this->height = obj.height;

}

double cyl::volume()

{

return M\_PI \*radius \* radius \* height;

}

double cyl::square()

{

return M\_PI \* radius \* radius \* 2 + height \* M\_PI \* radius;

}

cyl::~cyl()

{

std::cout << "Вызов деструктора " << this << std:: endl;

}

int main()

{

system("color f1");

setlocale(LC\_ALL, "RUS");

cyl a;

std::cout << "a - с параметрами по умолчанию (0, 0) Sq, Vol: " << a.square() <<", " << a.volume() << std::endl << std::endl;

int b\_r = 4;

int b\_h = 12;

cyl b(b\_r, b\_h);

std::cout << "b - с параметрами " << b\_r << ", " << b\_h << " Sq, Vol: " << b.square() << ", " << b.volume() << std::endl << std::endl;

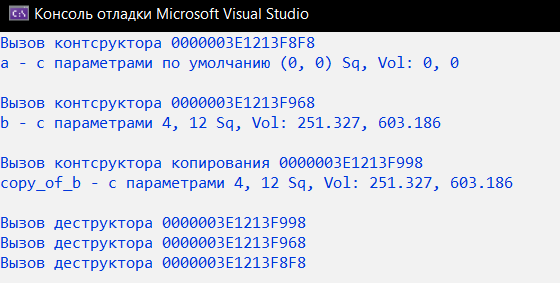
cyl copy\_of\_b(b);

std::cout << "copy\_of\_b - с параметрами " << b\_r << ", " << b\_h << " Sq, Vol: " << copy\_of\_b.square() << ", " << copy\_of\_b.volume() << std::endl << std::endl;

}

;

**Результат работы программы при исходных данных**



**Вывод**

Я закрепил особенности работы деструкторов, разных видов конструкторов и методов работы с классами.