



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ
КАФЕДРА

«Информатика и системы управления» (ИУ)
«Информационная безопасность» (ИУ8)

Лабораторная работа № 3
ПО КУРСУ
«Алгоритмические языки»
на тему «Изучение возможностей наследования классов»

Студент

ИУ8-25
(Группа)

В.В.Гоза
(И. О. Фамилия)

Преподаватель:

В. В. Соборова
(И.О. Фамилия)

2022 г.

Вариант 3

Условие:

Создать базовый класс «квадрат». Элементы класса: поле, задающее длину стороны (статус доступа `protected`); конструктор для инициализации поля; функция для вычисления площади квадрата; функция для печати поля и площади квадрата. Создать производный класс «куб». Элементы класса: конструктор для инициализации поля; переопределенная функция для вычисления объема куба (вместо площади) (внутри переопределенной функции должна вызываться функция из базового класса). Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Программа:

header.h:

```
#ifndef LABA3_HEADER_H
#define LABA3_HEADER_H
#pragma once
class Square {
protected:
    double a = 0;
public:
    Square();
    Square(double a);
    virtual double S();
    virtual void print();
};
class Cube: public Square {
public:
    Cube(double a);
    double S ();
    void print();
};
#endif //LABA3_HEADER_H
```

class.cpp:

```
#include <iostream>

#include "header.h"
#include <cmath>
Square::Square() {
    a = 0;
}
Square::Square(double n) {
    a = n;
}
double Square::S() {
    return pow(a, 2);
}
void Square::print() {
    std::cout << std::endl << "side: " << a << " area: " << S();
}
Cube::Cube(double n) {
    a = n;
}
double Cube::S() {
    return pow(a, 3);
}
void Cube::print() {
    std::cout << std::endl << "side: " << a << " volume: " << S();
}
```

main.cpp:

```
#include <iostream>
#include "cmake-build-debug/header.h"
int main() {
    double side;
    std::cout << "Enter the size of side: ";
    std::cin >> side;
    Square squ(side);
    double sidecube;
    std::cout << "\nEnter the size of side: ";
    std::cin >> sidecube;
    Cube cu(sidecube);
    Square *pr;
    pr = &squ;
    pr -> print();
    pr = &cu;
    pr -> print();
    return 0;
}
```