

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 «Информатика и системы управления»  
КАФЕДРА «Информационная безопасность»

ОТЧЕТ  
по лабораторной работе № 3  
по учебной дисциплине «Алгоритмические языки»  
на тему: «Изучение возможностей наследования классов»

Вариант 3

Выполнил:   
Студент 1 курса, гр. ИУ8-24  
Визгалов Ярослав

**Цель работы:**

Овладение навыками разработки программ на языке Си++, использующих возможности наследования классов для решения различных задач.

**Задачи работы:**

1. Изучить необходимые учебные материалы, посвященные наследованию классов в языке Си++
2. Разработать программу на языке Си++ для решения заданного варианта задания
3. Отладить программу
4. Представить результаты работы программы
5. Подготовить отчет по лабораторной работе

**Условие задачи:**

Создать базовый класс «вектор на плоскости». Элементы класса: поля, задающие координаты точки (статус доступа *protected*), определяющей конец вектора (начало вектора находится в точке с координатами 0, 0); конструктор для инициализации полей; функция для вычисления длины вектора, функция для печати полей и длины вектора. Создать производный класс «вектор в трехмерном пространстве». Элементы класса: дополнительное поле, задающее дополнительную координату; конструктор для инициализации полей; переопределенная функция для вычисления длины вектора; переопределенная функция для печати полей и длины вектора. Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

**Выполнение работы:**

#include <stdlib.h>  
#include <iostream>  
using namespace std;  
class Square // Базовый класс - "Квадрат"  
{  
protected:  
 double l; // Длина стороны  
public:  
 Square(double l): l(l) {} // Конструктор для инициализации поля  
  
 virtual double area() {  
 return l\*l;  
 }  
 void print() {  
 cout << "\nSide = " << l << " Area = " << l\*l << "\n";  
 }  
};  
class Cube: public Square // Производный класс - "Куб"  
{  
public:  
 Cube(double l): Square(l) {} // Конструктор  
  
 double area() override // Переопределенный метод area  
 {  
 return Square::area()\*l;  
 }  
};  
  
int main(int argc, char\* argv[])  
{  
 Square sq\_static(12);  
 cout << sq\_static.area() << "\n";  
 Cube c\_static(4);  
 cout << c\_static.area() << "\n";  
  
 Square\* sq\_dinamic = new Square(10);  
 cout << sq\_dinamic->area() << "\n";  
 Cube\* c\_dinamic = new Cube(7);  
 cout << c\_dinamic->area() << "\n";  
 system("pause"); // Останавливаем программу до нажатия любой клавиши  
 return 0;  
}

**Вывод:**

В результате выполнения лабораторной работы мы успешно создали иерархию классов квадрата и куба, а также реализовали методы для работы с объектами этих классов и продемонстрировали использование статического и динамического полиморфизма.