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»

15 Вариант

Выполнил:   
Студент 1 курса, гр. ИУ8-24  
Мирошниченко Максим

**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 <iostream>

#include <cmath>

using namespace std;

class Vector2D {

protected:

double x, y;

public:

Vector2D(double x = 0, double y = 0) : x(x), y(y) {}

double length() const {

return std::sqrt(x \* x + y \* y);

}

virtual void print() const {

cout << "Vector2D: (" << x << ", " << y << "), Length: " << length() << endl;

}

};

class Vector3D : public Vector2D {

private:

double z;

public:

Vector3D(double x = 0, double y = 0, double z = 0) : Vector2D(x, y), z(z) {}

double length() const {

return std::sqrt(x \* x + y \* y + z \* z);

}

void print() const override {

cout << "Vector3D: (" << x << ", " << y << ", " << z << "), Length: " << length() << endl;

}

};

int main() {

Vector2D\* vec2D = new Vector2D(3, 4);

vec2D->print(); // Вызов функции print() через указатель на базовый класс

Vector2D\* vec3D = new Vector3D(3, 4, 5);

vec3D->print(); // Вызов функции print() через указатель на базовый класс, но объект производного класса

return 0;

}

**Вывод:**

В результате выполнения лабораторной работы мы успешно создали базовый класс " сотрудник предприятия", реализовали методы для работы с объектами этих классов и продемонстрировали использование статического и динамического полиморфизма