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 9

Выполнил:   
Студент 1 курса, гр. ИУ8-24  
Катков Евгений

**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 «квадрат». Элементы класса: поле, задающее длину стороны (статус доступа *protected*); конструктор для инициализации поля; функция для вычисления периметра квадрата; функция для печати длины стороны и периметра. Создать производный класс «прямоугольник». Элементы класса: дополнительное поле, задающее другую сторону; конструктор для инициализации полей; переопределенная функция для вычисления периметра прямоугольника; переопределенная функция для печати длин сторон и периметра. 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>

using namespace std;

class Square {

protected:

double side; // Поле класса, хранящее длину стороны квадрата

public:

Square(double side) : side(side) {} // Конструктор класса Square для инициализации поля side

double getPerimeter() const { // Метод класса Square для вычисления периметра квадрата

return 4 \* side;

}

virtual void print() const { // Виртуальный метод для печати информации о квадрате

cout << endl;

cout << "Side length: " << side << endl; // Вывод длины стороны квадрата

cout << "Perimeter: " << getPerimeter() << endl; // Вывод периметра квадрата

}

};

class Rectangle : public Square {

protected:

double width; // Дополнительное поле класса Rectangle, задающее ширину прямоугольника

public:

Rectangle(double side, double width) : Square(side), width(width) {} // Конструктор класса Rectangle для инициализации полей

double getPerimeter() const { // Переопределенный метод для вычисления периметра прямоугольника

Square::getPerimeter(); // Вызов метода базового класса, но результат не используется

return 2 \* (side + width); // Вычисление периметра прямоугольника

}

void print() const { // Переопределенный метод для печати информации о прямоугольнике

Square::print(); // Вызов метода базового класса для вывода информации о квадрате

cout << "Width: " << width << endl; // Вывод ширины прямоугольника

cout << "Perimeter: " << getPerimeter() << endl; // Вывод периметра прямоугольника

}

};

int main() {

Square square(5); // Создание объекта квадрата

Rectangle rectangle(3, 6); // Создание объекта прямоугольника

square.print(); // Вывод информации о квадрате

cout << endl;

rectangle.print(); // Вывод информации о прямоугольнике

cout << endl;

Square\* s\_ptr = new Square(7); // Создание указателя на объект квадрата

Rectangle\* r\_ptr = new Rectangle(4, 8); // Создание указателя на объект прямоугольника

s\_ptr->print(); // Вывод информации о квадрате через указатель

cout << endl;

r\_ptr->print(); // Вывод информации о прямоугольнике через указатель

cout << endl;

return 0;

}

**Вывод:**

В результате выполнения лабораторной работы мы успешно создали иерархию классов "Квадрат" и "Прямоугольник", реализовали методы для работы с объектами этих классов и продемонстрировали использование статического и динамического полиморфизма.