ФАКУЛЬТЕТ «Информатика и системы управления»  
КАФЕДРА «Информационная безопасность»

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

Вариант 23

Выполнил:   
Студент 1 курса, гр. ИУ8-24  
Терехин Данила

**Цель работы:**

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

**Задачи работы:**

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

**Выполнение работы:**

**Main.cpp**

#include <iostream>

#include <string>

// Базовый класс "автомобиль"

class Car {

protected:

    std::string model;

    int maxSpeed;

public:

    Car(const std::string& model, int maxSpeed) : model(model), maxSpeed(maxSpeed) {}

    // Функция для печати параметров автомобиля

    void printParams() const {

        std::cout << "Model: " << model << ", Max Speed: " << maxSpeed << " km/h" << std::endl;

    }

    // Виртуальная функция для динамического полиморфизма

    virtual void printParamsDynamic() const {

        std::cout << "Dynamic - Model: " << model << ", Max Speed: " << maxSpeed << " km/h" << std::endl;

    }

};

// Производный класс "автобус"

class Bus : public Car {

private:

    int maxPassengers;

public:

    Bus(const std::string& model, int maxSpeed, int maxPassengers)

        : Car(model, maxSpeed), maxPassengers(maxPassengers) {}

    // Переопределенная функция печати параметров автобуса (статический полиморфизм)

    void printParams() const {

        Car::printParams(); // Вызов функции из базового класса

        std::cout << "Max Passengers: " << maxPassengers << std::endl;

    }

    // Переопределенная виртуальная функция для динамического полиморфизма

    void printParamsDynamic() const override {

        Car::printParamsDynamic(); // Вызов функции из базового класса

        std::cout << "Dynamic - Max Passengers: " << maxPassengers << std::endl;

    }

};

int main() {

    // Создание объекта базового класса

    Car car("Toyota Camry", 220);

    car.printParams();

    car.printParamsDynamic();

    // Создание объекта производного класса

    Bus bus("Mercedes-Benz Tourismo", 180, 50);

    bus.printParams();

    bus.printParamsDynamic();

    // Демонстрация динамического полиморфизма

    Car\* carPtr = &bus;

    carPtr->printParamsDynamic(); // Вызов переопределенной функции из производного класса

    return 0;

}

**Результат работы:**Model: Toyota Camry, Max Speed: 220 km/h

Dynamic - Model: Toyota Camry, Max Speed: 220 km/h

Model: Mercedes-Benz Tourismo, Max Speed: 180 km/h

Max Passengers: 50

Dynamic - Model: Mercedes-Benz Tourismo, Max Speed: 180 km/h

Dynamic - Max Passengers: 50

Dynamic - Model: Mercedes-Benz Tourismo, Max Speed: 180 km/h

Dynamic - Max Passengers: 50

**Вывод:**

В результате выполнения данной лабораторной работы я успешно овладел навыками разработки программ на языке Си++, использующих наследование классов и динамический полиморфизм. Пройдя через все этапы задачи, я углубил свои знания в области объектно-ориентированного программирования и получил практический опыт в реализации наследования и переопределения виртуальных функций. Этот опыт укрепил мою способность создавать гибкие и расширяемые программные системы, что является ключевым аспектом профессионального программирования.