Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ

УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Автоматизированных систем управления (АСУ)

Классы. Открытые и закрытые уровни доступа. Конструкторы. Инициализация данных объекта. Определение методов. Создание объекта в памяти. Стандартные потоки ввода-вывода.

ООП  
Лабораторная работа 1  
Вариант 14

Выполнила:  
Студентка гр. З-433П8-5  
Медведева Юлия Евгеньевна  
«14» марта 2025г.

Проверил:  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
(должность, ученая степень, звание)  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Косова Анастасия Евгеньевна/  
(подпись)  
«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_\_г.

Томск 2025

# Оглавление

1. Введение
2. Задание
3. 1 Теоретическая часть
4. 1.1 Классы C++
5. 1.2 Уровни доступа в классах
6. 1.3 Определение методов
7. 1.4 Инициализация данных объекта
8. 1.5 Конструкторы и деструкторы
9. 1.6 Наследование
10. 1.7 Проверка вводимых данных и сеттеры
11. 2 Описание хода выполнения работы
12. Заключение
13. Список литературы
14. Приложение 1

# Введение

Цель: изучить объектно-ориентированный подход в C++ на примере работы с классами, конструкторами, деструкторами, методами и наследованием.  
Задачи:  
• Ознакомиться с принципами инкапсуляции, наследования и полиморфизма.  
• Реализовать класс Car и его наследник AdditionalCar.  
• Продемонстрировать работу всех конструкторов (по умолчанию и с параметрами).  
• Добавить методы проверки данных через сеттеры.  
• Реализовать вывод сообщений в деструкторах для подтверждения их вызова.

# Задание

Предметная область: автомобиль. Необходимо реализовать программу, содержащую базовый класс Car с характеристиками — марка, номер, цена. Создать производный класс AdditionalCar, добавляющий описание. Реализовать четыре варианта создания объектов (статически и динамически).

# 1 Теоретическая часть

Классы в C++ — это основа объектно-ориентированного программирования. Они объединяют данные и функции, обеспечивая инкапсуляцию и удобное управление логикой программы.

Конструкторы и деструкторы позволяют управлять созданием и уничтожением объектов. При этом деструкторы могут использоваться для вывода диагностической информации, подтверждающей корректную работу программы.

Наследование используется для расширения базового класса новыми свойствами и методами. В данной работе класс AdditionalCar наследует базовый класс Car и добавляет описание автомобиля.

Методы проверки данных через сеттеры позволяют обеспечить корректность входных данных и защитить внутреннее состояние объекта.

# 2 Описание хода выполнения работы

В ходе работы был реализован класс Car с полями brand\_, number\_ и price\_. Для каждого поля созданы методы чтения и записи с проверкой допустимости вводимых данных. Также был создан класс-наследник AdditionalCar, добавляющий поле mainInfo\_ для хранения описания автомобиля. Деструкторы обоих классов содержат сообщения о вызове, что позволяет проследить их работу.

Программа демонстрирует четыре варианта создания объектов:   
1) статическое создание с параметрами по умолчанию,  
2) статическое создание с параметрами и наследованием,  
3) динамическое создание с параметрами по умолчанию,  
4) динамическое создание с параметрами.  
При вводе данных реализована проверка корректности: цена и номер не могут быть отрицательными, а марка машины должна содержать хотя бы одну букву.

# Заключение

В ходе лабораторной работы были закреплены знания об объектно-ориентированном программировании в C++. Были реализованы базовые и производные классы, конструкторы, деструкторы, а также методы проверки данных. Программа корректно демонстрирует работу механизмов инкапсуляции и наследования.

# Список литературы

1. Страуструп Б. Язык программирования C++. — 4-е изд. — Бином, 2022.  
2. Дейтел П. Дейтел. Как программировать на C++. — М.: Бином-Пресс, 2008.  
3. Липпман С., Лажой Ж., Муллиган Б. Э. Язык программирования C++. Базовый курс. — Диалектика, 2018.

# Приложение 1. Листинг программы

Файл main.h и main.cpp представлены в финальной версии лабораторной работы.