Министерство цифрового развития, связи и массовых коммуникаций
Российской Федерации Сибирский государственный университет телекоммуникаций и информатики

кафедра ТС и ВС

Лабораторная работа 1

по дисциплине

«Объектно-ориентированное программирование»

Выполнил: Демин С. А.
Группа: ИКС-433
Вариант: 5

Проверил: Нейдорф П. Я.

Новосибирск, 2025

Классы

Цель работы: ознакомиться с понятиями класса и объекта, научиться создавать простые классы, их поля и методы.

Задание: Создать собственный класс согласно варианту задания. Создаваемый класс должен содержать несколько полей (по выбору студента), типы которых соответствуют варианту задания (можете ввести дополнительные поля), конструктор для инициализации значений полей объекта, методы для изменения значений полей объекта и для получения этих значений, кроме того метод для вывода всей имеющейся информации об объекте. Метод main() должен демонстрировать работу всех методов объекта и значения всех полей до и после изменения.

Вариант:

№ варианта	Суперкласс	Поля
5	Фирма-производитель	Страна, тип оборудования

Текст класса:

```
class Manufacturer {
   private:
   string country;
   string type;
   int year;
   string segment;

public:
Manufacturer(string c, string e, int y, string m) {
   country = c;
   type = e;
   year = y;
   segment = m;}

   string getCountry() { return country; }
   string getType() { return type; }
   int getYear() { return year; }
   string getSegment() { return segment; }

   void setCountry(string c) { country = c; }
```

```
    void setType(string e) { type = e; }
    void setYear(int y) { year = y; }
    void setSegment(string m) { segment = m; }

    void outputInfo() {
        cout<< "Страна: " << country << endl;
        cout<< "Тип оборудования: " << type << endl;
        cout<< "Год основания: " << year << endl;
        cout<< "Рыночный сегмент: " << segment << endl;
        cout<< "----------------------------" << endl;}};
```

Результаты:

ПЕРВОНАЧАЛЬНЫЕ ДАННЫЕ:

Страна: Германия

Тип оборудования: Промышленные станки

Год основания: 1886

Рыночный сегмент: Премиум

Страна: Япония

Тип оборудования: Автомобили

Год основания: 1937

Рыночный сегмент: Средний

Страна: США

Тип оборудования: Сетевое оборудование

Год основания: 1984

Рыночный сегмент: Средний

ПОСЛЕ ИЗМЕНЕНИЙ:

Страна: Германия

Тип оборудования: Промышленные станки

Год основания: 2024

Рыночный сегмент: Люкс

Страна: Япония

Тип оборудования: Автомобили

Год основания: 1937

Рыночный сегмент: Средний

Страна: США

Тип оборудования: Сетевое оборудование

Год основания: 1984

Рыночный сегмент: Средний

Года основания

Bosch: 2024

toyota: 19s

cisco: 1984

Описание результатов:

Сначала в данных были три компании: немецкая Bosch (промышленные станки, основана в 1886 году, сегмент "Премиум"), японская Toyota (автомобили, 1937 год, "Средний") и американская Cisco (сетевое оборудование, 1984 год, "Средний"). После изменений данные двух компаний — японской и американской — остались прежними. Однако информация о немецкой компании была полностью обновлена: теперь она указана как основанная в **2024** году и работающая в сегменте **"Люкс"**.

Выводы:

В ходе выполнения лабораторной работы был успешно реализован класс **Manufacturer** (Фирма-производитель) согласно варианту №5. Были изучены и применены на практике основные принципы объектно-ориентированного программирования:

1. **Создание класса** с приватными полями: country (страна), type (тип оборудования), year (год основания) и segment (рыночный сегмент)
2. **Реализован конструктор** для инициализации объектов при их создании
3. **Разработаны методы доступа** (get/set) для каждого поля, обеспечивающие инкапсуляцию данных
4. **Создан метод outputInfo()** для вывода полной информации об объекте в консоль

В методе main() была продемонстрирована работа всех методов класса:

- Создание трех объектов фирм (Bosch, Toyota и Cisco) с исходными данными
- Вывод первоначальной информации о фирмах
- Изменение значений полей с использованием set-методов
- Вывод обновленной информации после изменений

- Демонстрация работы get-методов для получения отдельных значений

Программа выполнена без ошибок компиляции и выполнения. Результаты работы полностью соответствуют ожидаемым данным, что подтверждает корректность реализации класса и его методов.

Контрольные вопросы:

1. Объясните понятие класса.

Класс — это определяемый пользователем тип данных, который выступает в качестве шаблона или чертежа для создания объектов. Он инкапсулирует данные (поля) и функции (методы), которые работают с этими данными, в одну логическую единицу.

2. Объясните понятие объекта.

Объект — это экземпляр (конкретная реализация) класса. Если класс — это чертеж, то объект — это дом, построенный по этому чертежу. Объект занимает место в памяти и содержит конкретные значения полей, определённых в классе.

3. Что представляют собой поля и методы класса.

- **Поля класса** — это переменные, объявленные внутри класса. Они хранят состояние или данные объекта.
- **Методы класса** — это функции, объявленные внутри класса. Они определяют поведение объекта, то есть операции, которые можно выполнять с его данными.

4. Что такое статические поля и методы?

Статические поля и методы принадлежат самому классу, а не его отдельным объектам. Они существуют в единственном экземпляре для всего класса и могут быть использованы без создания объекта.

5. Охарактеризуйте модификаторы доступа.

Модификаторы доступа определяют видимость полей и методов класса:

- **private**: доступ разрешён только внутри самого класса.
- **public**: доступ разрешён из любого места программы.
- **protected**: доступ разрешён внутри класса и его производных классов.

6. В чём заключается преимущество объектно-ориентированного программирования?

Преимущество ООП заключается в возможности моделировать реальные сущности, повышать переиспользуемость кода,

структурировать программу, упрощать её поддержку и модификацию за счёт инкапсуляции, наследования и полиморфизма.

Весь код:

```
#include <iostream>
#include <string>
using namespace std;

class Manufacturer {
   private:
   string country;
   string type;
   int year;
   string segment;

public:
Manufacturer(string c, string e, int y, string m) {
   country = c;
   type = e;
   year = y;
   segment = m;}

   string getCountry() { return country; }
   string getType() { return type; }
   int getYear() { return year; }
   string getSegment() { return segment; }

   void setCountry(string c) { country = c; }
   void setType(string e) { type = e; }
   void setYear(int y) { year = y; }
   void setSegment(string m) { segment = m; }

   void outputInfo() {
      cout<< "Страна: " << country << endl;
      cout<< "Тип оборудования: " << type << endl;
      cout<< "Год основания: " << year << endl;
      cout<< "Рыночный сегмент: " << segment << endl;
```

```
        cout<< "----------------------------" << endl;}};

    int main() {
        Manufacturer bosch("Германия", "Промышленные станки", 1886,
"Премиум");
        Manufacturer toyota("Япония", "Автомобили", 1937, "Средний");
        Manufacturer cisco("США","Сетевое оборудование",1984,"Средний");

        cout << "ПЕРВОНАЧАЛЬНЫЕ ДАННЫЕ:" << endl;
        bosch.outputInfo();
        toyota.outputInfo();
        cisco.outputInfo();

        bosch.setSegment("Люкс");
        bosch.setYear(2024);

        cout<< "ПОСЛЕ ИЗМЕНЕНИЙ:" << endl;
        bosch.outputInfo();
        toyota.outputInfo();
        cisco.outputInfo();

        cout<< "Года основания" << endl;
        cout << "Bosch: "<<bosch.getYear()<< endl;
        cout<<"toyota: "<<toyota.getYear()<<endl;
        cout<< "cisco: "<<cisco.getYear()<<endl;
    }
```