

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ им. В. Г. ШУХОВА» (БГТУ им. В.Г. Шухова)

Кафедра программного обеспечения
вычислительной техники и автоматизированных
систем

Лабораторная работа №5

по дисциплине: ООП

тема: «Классы, виды отношений. Наследование»

Выполнил: студент группы ВТ-231

Масленников Д. А.

Проверили:
Буханов Д. Г.

Белгород 2025

Цель работы: Получение теоретических знаний в области разработки классов, получение практических навыков реализаций классов и отношений между ними.

Задание к работе:

Задание 1. В соответствии с вариантом $((\text{номер по списку} + 5) \% 10) + 1$ выполнить построение объектной модели (использовать не менее 5 объектов) заданной предметной области и разработать диаграмму классов для описанной объектной модели (не менее 7 классов).

Задание 2. Реализовать некоторые предложенные классы из задания 1, соответствующие постановке задачи задания 2.

Вариант 10

Задание 1

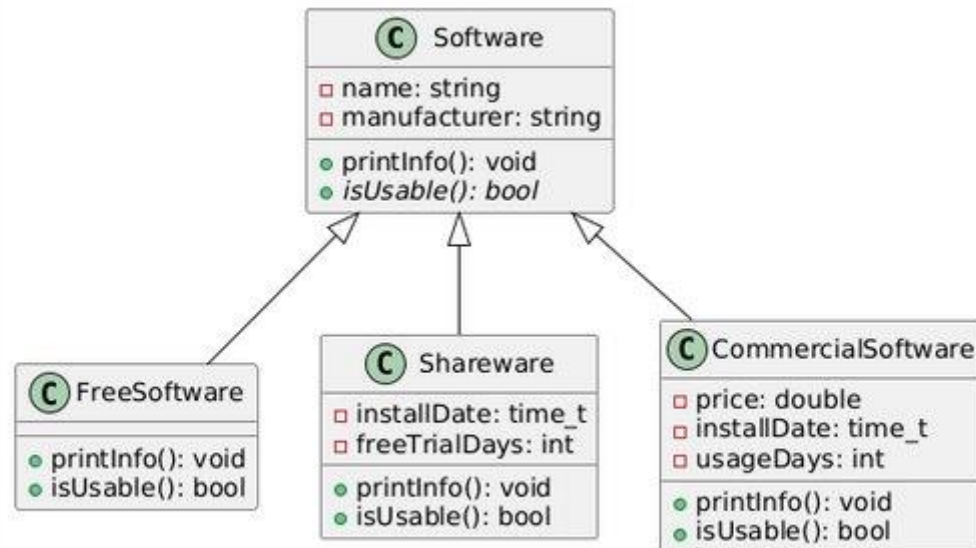
Справочник зарегистрированных программных продуктах.

Задание 2

1. Создать абстрактный класс Программное_обеспечение с методами, позволяющими вывести на экран информацию о программном обеспечении, а также определить соответствие возможности использования (на момент текущей даты).
2. Создать производные классы: Свободное (название, производитель), Условно-бесплатное (название, производитель, дата установки, срок бесплатного использования), Коммерческое (название, производитель, цена, дата установки, срок использования) со своими методами вывода информации на экран, и определения возможности использования на текущую дату.
3. Создать базу (массив) из n видов программного обеспечения, вывести полную информацию из базы на экран, а также организовать поиск программного обеспечения, которое допустимо использовать на текущую дату.

Задание 1:

Справочник зарегистрированных программных продуктов.



Задание 2:

```
#include <iostream>

#include <ctime>
#include <string>

using namespace std;

class Software {
protected:
    string name;
    string manufacturer;
public:
    Software(const string& name, const string& manufacturer)
        : name(name), manufacturer(manufacturer) {}

    virtual void printInfo() const {
        cout << "Программное обеспечение: " << name << " от " << manufacturer << endl;
    }

    virtual bool isUsable() const = 0;
};

class FreeSoftware : public Software {
public:
    FreeSoftware(const string& name, const string& manufacturer)
        : Software(name, manufacturer) {}

    void printInfo() const override {
        cout << "Бесплатное программное обеспечение: " << name << " от " << manufacturer << endl;
    }

    bool isUsable() const override {
        return true;
    }
};

class Shareware : public Software {
private:
    time_t installDate;
    int freeTrialDays;
public:
    Shareware(const string& name, const string& manufacturer, time_t installDate, int freeTrialDays)
        : Software(name, manufacturer), installDate(installDate), freeTrialDays(freeTrialDays) {}

    void printInfo() const override {
```

```

        cout << "Условно-бесплатное программное обеспечение: " << name << " от " << manufacturer << endl;
        cout << "Дата установки: " << ctime(&installDate);
        cout << "Дни бесплатного использования: " << freeTrialDays << endl;
    }

    bool isUsable() const override {
        time_t now = time(0);
        return difftime(now, installDate) < freeTrialDays * 86400;
    }
};

class CommercialSoftware : public Software {
private:
    double price;
    time_t installDate;
    int usageDays;
public:
    CommercialSoftware(const string& name, const string& manufacturer, double price, time_t installDate, int
usageDays)
        : Software(name, manufacturer), price(price), installDate(installDate), usageDays(usageDays) {}

    void printInfo() const override {
        cout << "Коммерческое программное обеспечение: " << name << " от " << manufacturer << endl;
        cout << "Цена: $" << price << endl;
        cout << "Дата установки: " << ctime(&installDate);
        cout << "Дни использования: " << usageDays << endl;
    }

    bool isUsable() const override {
        time_t now = time(0);
        return difftime(now, installDate) < usageDays * 86400;
    }
};

int main() {
    const int n = 3;
    Software* softwareList[n];

    softwareList[0] = new FreeSoftware("Linux", "GNU Project");
    softwareList[1] = new Shareware("WinRAR", "RARLAB", time(0), 30);
    softwareList[2] = new CommercialSoftware("Microsoft Office", "Microsoft", 200, time(0), 365);

    cout << "Полный список программного обеспечения:" << endl;
    for (int i = 0; i < n; ++i) {
        softwareList[i]->printInfo();
        cout << "-----" << endl;
    }
}

```

```

cout << "Доступное программное обеспечение:" << endl;
for (int i = 0; i < n; ++i) {
    if (softwareList[i]->isUsable()) {
        softwareList[i]->printInfo();
        cout << "-----" << endl;
    }
}

return 0;
}

```

```

• > g++ -o res main.cpp
• > ./res
Полный список программного обеспечения:
Бесплатное программное обеспечение: Linux от GNU Project
-----
Условно-бесплатное программное обеспечение: WinRAR от RARLAB
Дата установки: Thu Mar 20 01:30:19 2025
Дни бесплатного использования: 30
-----
Коммерческое программное обеспечение: Microsoft Office от Microsoft
Цена: $200
Дата установки: Thu Mar 20 01:30:19 2025
Дни использования: 365
-----
Доступное программное обеспечение:
Бесплатное программное обеспечение: Linux от GNU Project
-----
Условно-бесплатное программное обеспечение: WinRAR от RARLAB
Дата установки: Thu Mar 20 01:30:19 2025
Дни бесплатного использования: 30
-----
Коммерческое программное обеспечение: Microsoft Office от Microsoft
Цена: $200
Дата установки: Thu Mar 20 01:30:19 2025
Дни использования: 365
-----

```

Вывод: получил теоретические и практические навыки в области разработки классов и отношений между ними