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:**

**Задание1.** В соответствии с вариантом (((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, соответствующиепостановке задачи задания 2.

**Вариант 10**

Задание 1

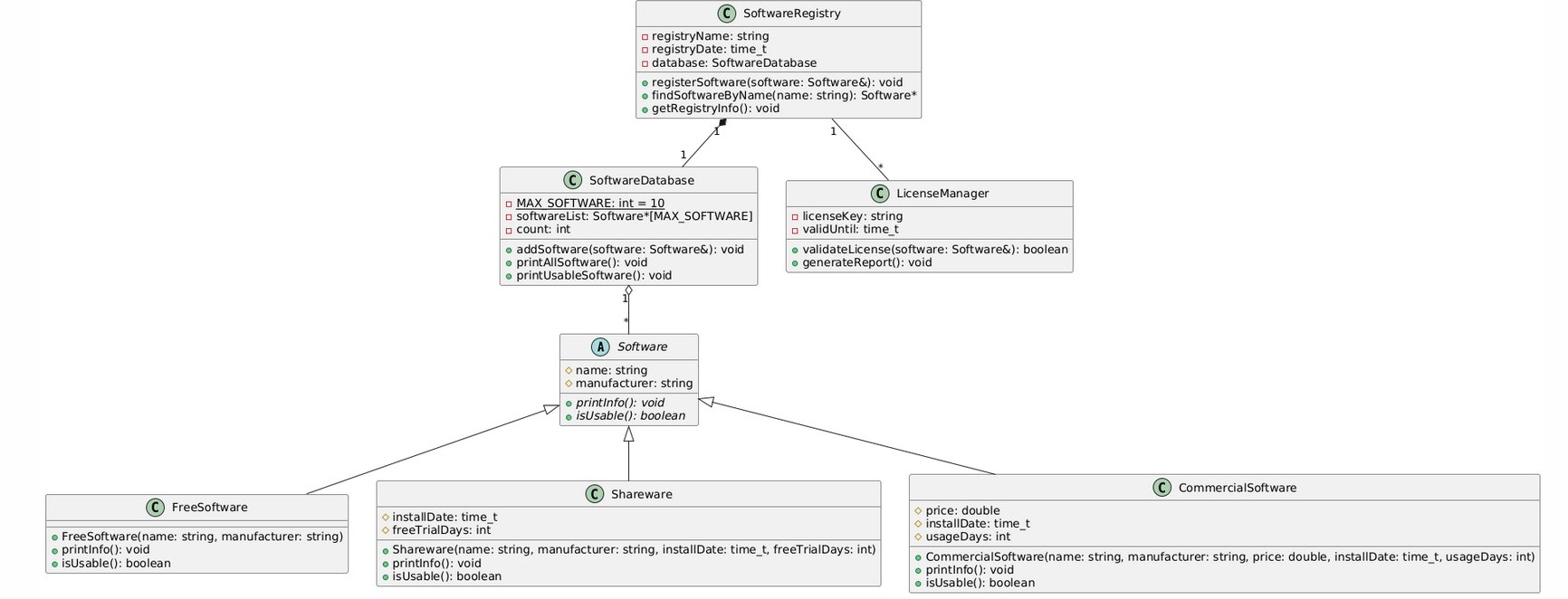
Справочник зарегистрированных программных продуктах.

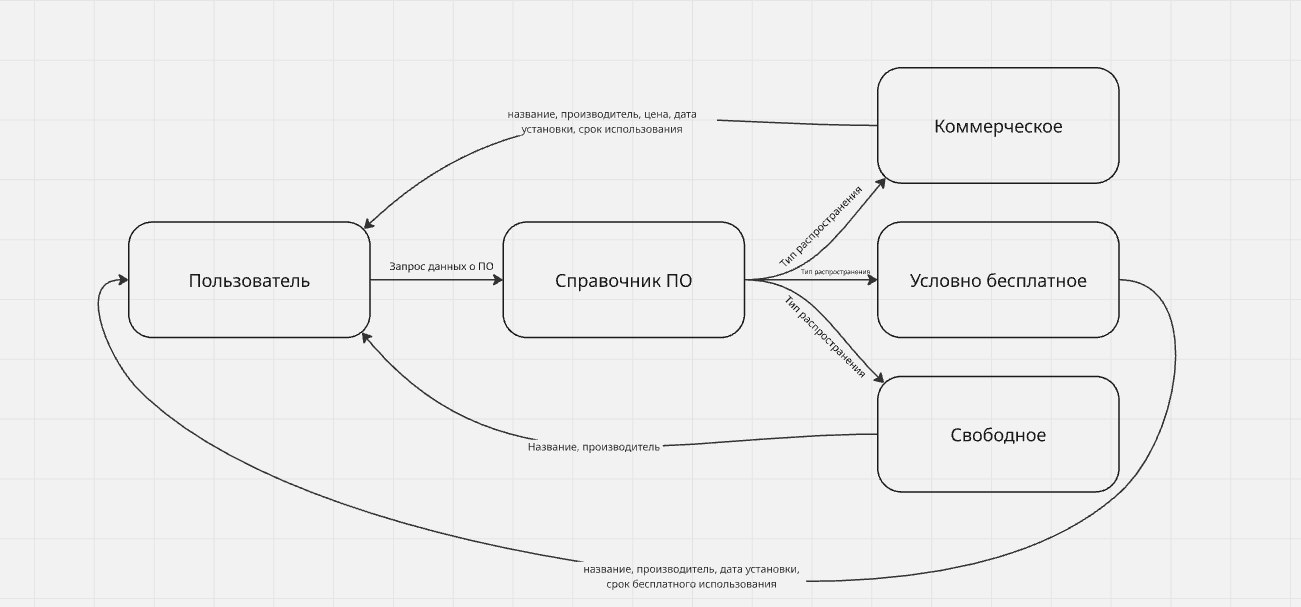
Задание 2

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 продуктах.**

****

****

**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 = 0;**

**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;**

**}**

**};**

**class SoftwareDatabase {**

**private:**

**static const int MAX\_SOFTWARE = 10;**

**Software\* softwareList[MAX\_SOFTWARE];**

**int count = 0;**

**public:**

**void addSoftware(Software& software) {**

**if (count < MAX\_SOFTWARE) {**

**softwareList[count++] = &software;**

**}**

**}**

**void printAllSoftware() const {**

**cout << "Полный список программного обеспечения" << endl;**

**for (int i = 0; i < count; ++i) {**

**softwareList[i]->printInfo();**

**cout << "-------------------------" << endl;**

**}**

**}**

**void printUsableSoftware() const {**

**cout << "Доступное программное обеспечение:" << endl;**

**for (int i = 0; i < count; ++i) {**

**if (softwareList[i]->isUsable()) {**

**softwareList[i]->printInfo();**

**cout << "-------------------------" << endl;**

**}**

**}**

**}**

**};**

**int main() {**

**FreeSoftware linuxOS("Linux", "GNU Project");**

**Shareware winrar("WinRAR", "RARLAB", time(0), 30);**

**CommercialSoftware office("Microsoft Office", "Microsoft", 200, time(0), 365);**

**SoftwareDatabase database;**

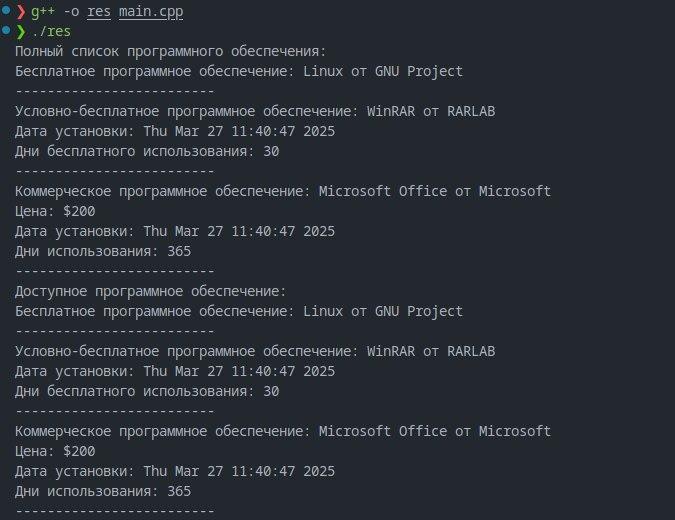
**database.addSoftware(linuxOS);**

**database.addSoftware(winrar);**

**database.addSoftware(office);**

**database.printAllSoftware();**

**database.printUsableSoftware();**

**return 0;****}**

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