МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ

Ордена Трудового Красного Знамени

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Московский технический университет связи и информатики»**

Кафедра «Информационные технологии»

Информационные технологии и программирование

Лабораторная работа №2

Выполнил: Студент группы

БВТ2207

Кривенок Вадим

Москва

2023

**Цель работы:** в данной лабораторной работе необходимо использовать классы, чтобы описать, как эти объекты работают.

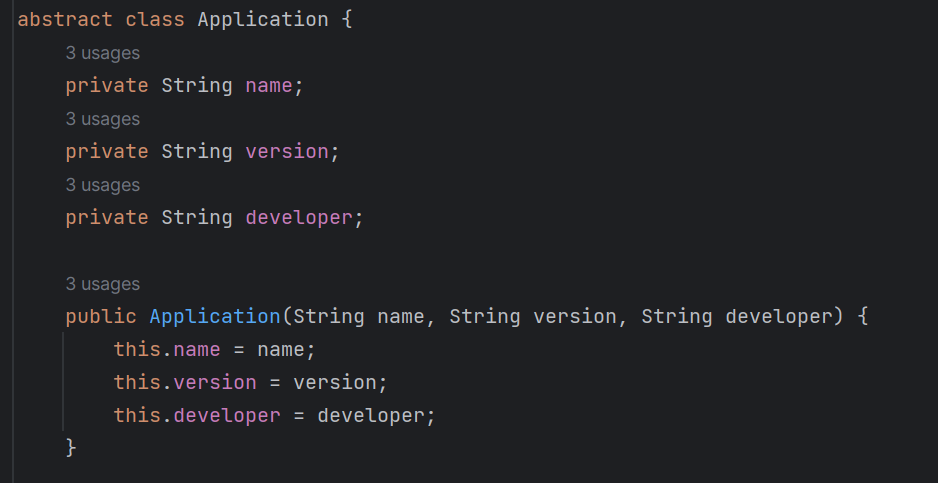
**Выполнение.**

Создайте иерархию классов. Ваша иерархия должна содержать:

* абстрактный класс - 2 уровня наследуемых классов (классы должны содержать в себе минимум 3 поля и 2 метода, описывающих поведение объекта)
* демонстрацию реализации всех принципов ООП (абстракция, модификаторы доступа, перегрузка, переопределение)
* наличие конструкторов (в том числе по умолчанию)
* наличие геттеров и сеттеров
* ввод/вывод информации о создаваемых объектах
* предусмотрите в одном из классов создание счетчика созданных объектов с использованием статической переменной, продемонстрируйте работ

Вариант №12:

* Базовый класс: Приложение
* Дочерние классы: Социальная сеть, Игра, Погода

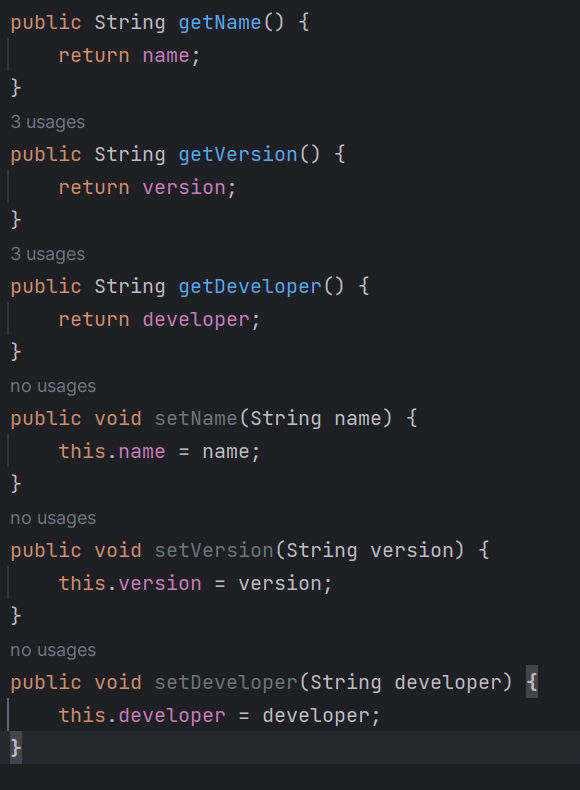
****Рис.1 – абстрактный класс Приложение.

Сначала создаем абстрактный класс "Application" (приложение).

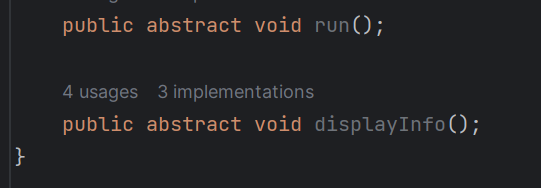
Объявляем приватные переменные "name", "version" и "developer".

Создаём конструктор класса "Application" с параметрами "name", "version" и "developer".

Внутри конструктора устанавливаются значения переданных параметров в соответствующие переменные класса.

Рис.2 – геттеры и сеттеры.

Эти методы позволяют получать значения и изменять значения переменных `name`, `version` и `developer` объекта класса ` Application ` извне класса. Геттеры возвращают текущие значения переменных, а сеттеры устанавливают новые значения переменных на основе переданных параметров.

****Рис.3 – абстрактные методы для описания поведения приложения.

Объявлены два абстрактных метода: run() и displayInfo(). Абстрактный метод не содержит реализации внутри абстрактного класса, но должен быть реализован в каждом классе-наследнике.

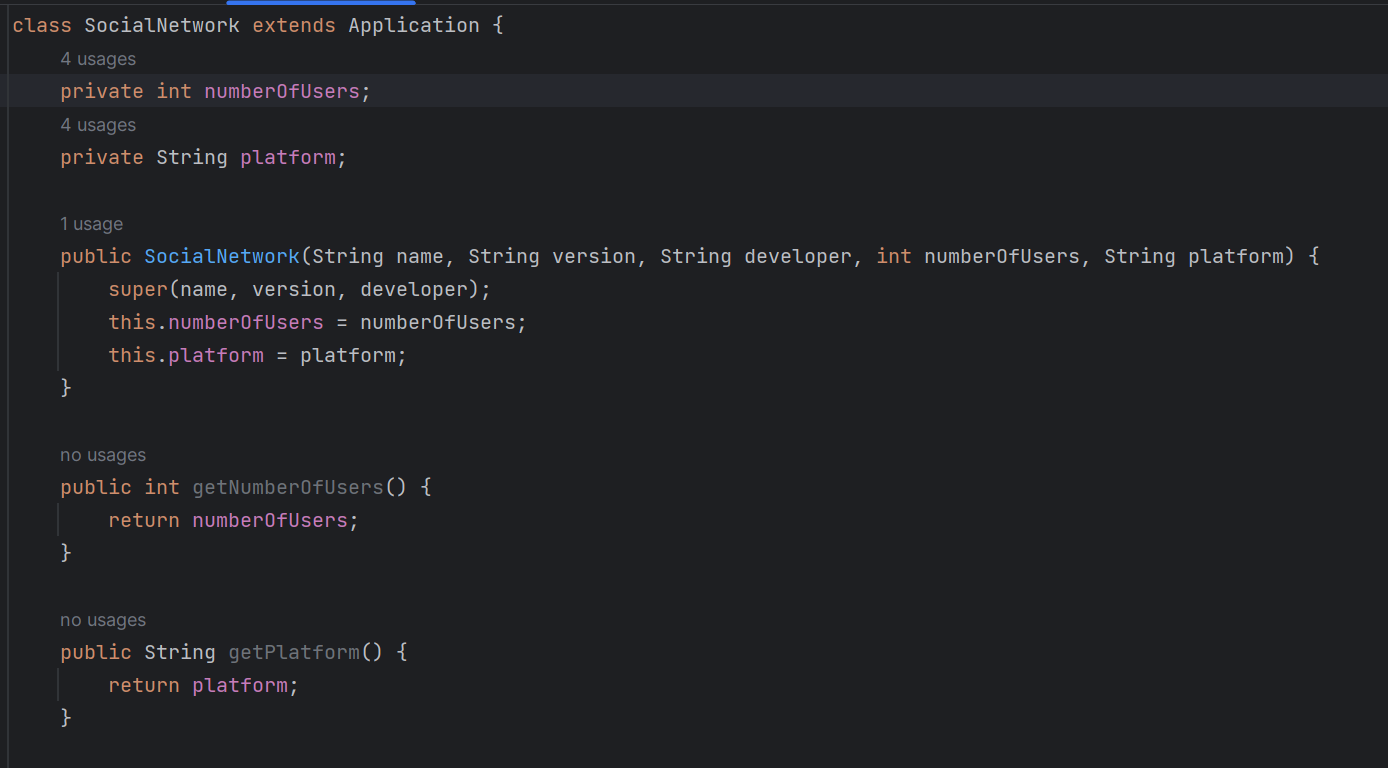
Рис.4(1) – класс социальная сеть.

Рис.4(2) – класс социальная сеть.

Класс SocialNetwork наследует все переменные и методы, определенные в классе Application.

private int numberOfUsers: Объявление приватной переменной numberOfUsers типа int, которая указывает количество пользователей в социальной сети.

private String platform: Объявление приватной переменной platform типа String, которая указывает платформу, на которой работает социальная сеть.

public SocialNetwork(String name, String version, String developer, int numberOfUsers, String platform): Создание конструктора класса SocialNetwork с параметрами name, version, developer, numberOfUsers и platform. Внутри конструктора вызывается конструктор суперкласса Application с параметрами name, version и developer, чтобы установить значения этих переменных. Затем значения параметров numberOfUsers и platform устанавливаются для соответствующих переменных класса SocialNetwork.

@Override: Аннотация, указывающая, что следующий метод переопределяет метод суперкласса.

public void run(): Переопределение метода run() из суперкласса Application. В этом методе выводится сообщение о том, что социальная сеть запущена, используя методы getName() для получения имени приложения.

public void displayInfo(): Переопределение метода displayInfo() из суперкласса Application. В этом методе выводится информация о приложении, включая его имя, версию, разработчика, количество пользователей и платформу, используя геттеры для получения соответствующих значений.

По аналогии прописываем и другие классы.

Рис.5(1) – класс игра.

Рис.5(2) – класс игра.

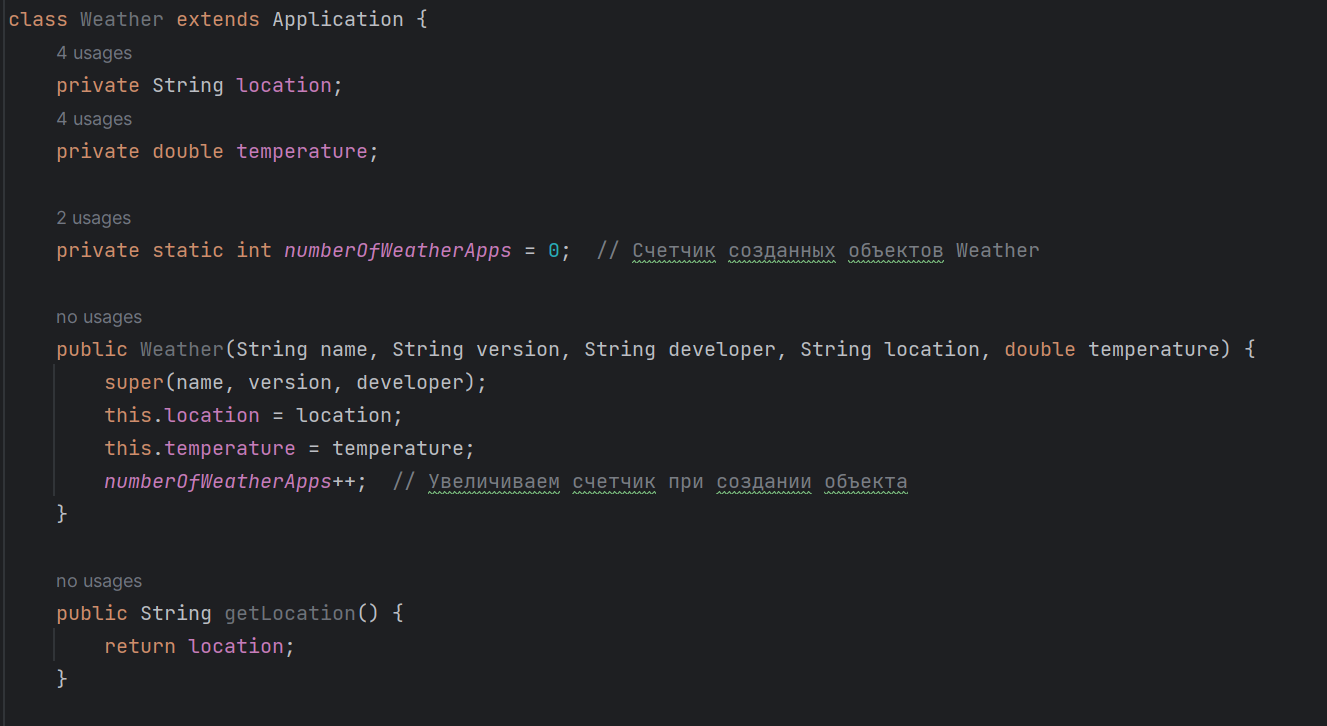
Рис.6(1) – класс погода.

Рис.6(2) – класс погода.

Рис.6(3) – класс погода.

**Вывод.**

В данной лабораторной работе я использовал классы, чтобы описать, как эти объекты работают.