**Лабораторная работа №7**

**РАЗРАБОТКА ПРИЛОЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ**

**ООП**

**Вариант 9.**

**Цель работы:** ознакомиться с методологией объектно-ориентированного программирования, изучить реализацию данной методологии в языке Python.

**Краткая теория**

Объектно-ориентированное программирование (ООП) – это одна из методологий (парадигм) программирования (структурное, функциональное), где компоненты программы представляются в виде объектов. Объект – это экземпляр класса, имеющего поля и методы. Большинство современных языков программирования, таких как Java, C++, C# и другие, реализуют данный подход, в том числе и Python. В данной лабораторной работе не рассматривается теория ООП, а только реализация данной методологии в Python 3.

В Python существуют три вида модификаторов доступа: public, protected, private. Модификатор доступа public разрешает доступ к переменным из любой точки вне и внутри класса. Модификатор доступа private допускает обращения к переменным только внутри класса. Модификатор доступа protected разрешает доступ из любой точки вне и внутри класса, внутри пакета, а также в классах-наследниках.

Инкапсуляция – это сокрытие данных одного класса от прямого обращения к его данным другими классами. Суть заключается в том, чтобы предоставлять доступ к переменным класса через методы геттеры и сеттеры, в которых может происходить логика, связанная с работой защищаемой переменной.

Наследование – это предоставление характеристик класса родителя классу-потомку. Данный механизм позволяет в несколько раз сократить количество кода за счет его повторного использования. Стоит также отметить, что в Python существует поддержка множественного наследования.

Полиморфизм заключается в способности объекта вести себя по-разному. Под полиморфизмом рассмотрим перегрузку и переопределение методов.

Перегрузка метода – это способность метода вести себя по-разному при определенном наборе параметров. Переопределение метода заключается в изменении логики метода с таким же наименованием и количеством параметров, что и метод в классе-родителе.

**Задание. Вариант 9.**

Написать класс Publisher и несколько различных классов с постфиксом Subscriber. Реализовать между данными классами отношения по паттерну Observer.

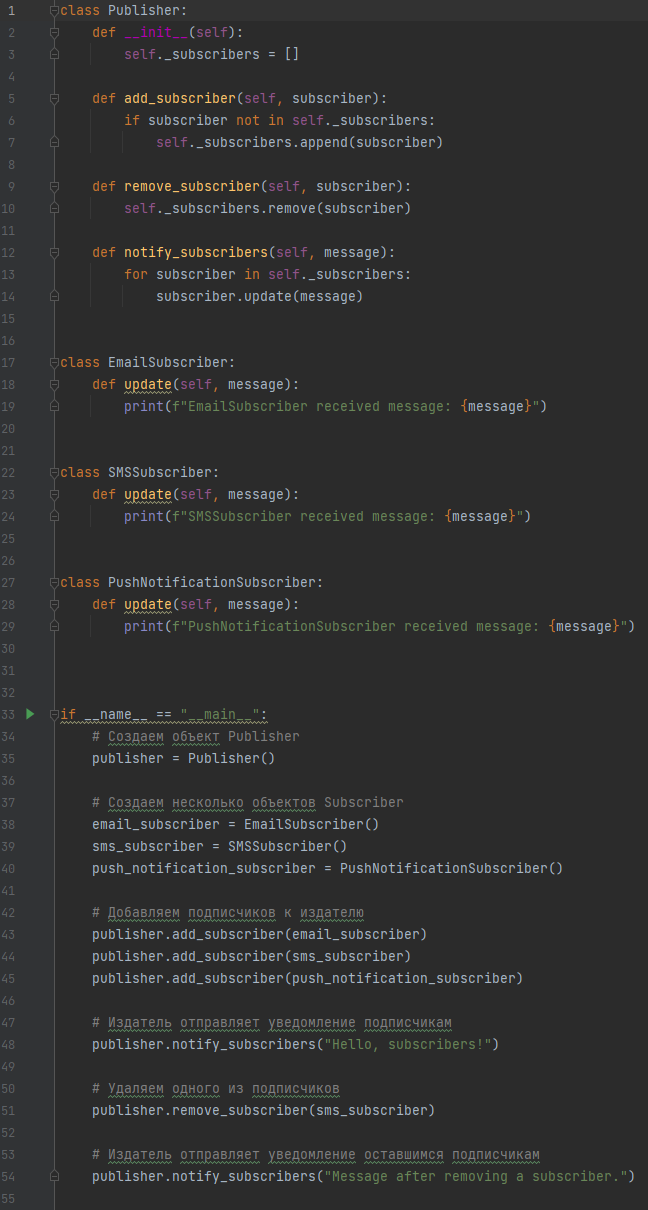


Рисунок 7.1 - код программы



Рисунок 7.2 - результат работы

**Вывод:** в ходе выполнения лабораторной работы была изучена методология объектно-ориентированного программирования, изучена реализация данной методологии в языке Python.

|  |  |
| --- | --- |
| Выполнил | Нестеренко М.С. |
| Проверил | Елкин Н.С. |