**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА**   
**И ГОСУДАРСТВЕННОЙ СЛУЖБЫ**

**ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

**НИЖЕГОРОДСКИЙ ИНСТИТУТ УПРАВЛЕНИЯ – филиал РАНХиГС**

**Отчет по программированию №2**

Выполнил: студент группы Иб-321

Комышков Владислав Дмитриевич

**Задание 1**

**Текст задания**

Задание по проекту «Подбрасывание\_Монет»:

1. Создать класс-имитатор подбрасывания монеты в отдельном модуле.
2. Пошагово описать с графическим отображением действия методов в основной программе и классов в скрипте «Подбрасывание монеты».
3. Обеспечить подбрасывание монеты 10 раз!
4. Проверить, при каком числе срабатывания проявится равновероятность исходов подбрасывания «идеальной» монеты.
5. Смоделировать подбрасывание несколько раз нескольких монет одновременно. Каждый экземпляр имеет собственный атрибут начального положения монеты, установленный случайным образом.

**Ход работы**

Создаем класс Coin (Рис 1.1). Он будет имитировать подбрасывание монеты.

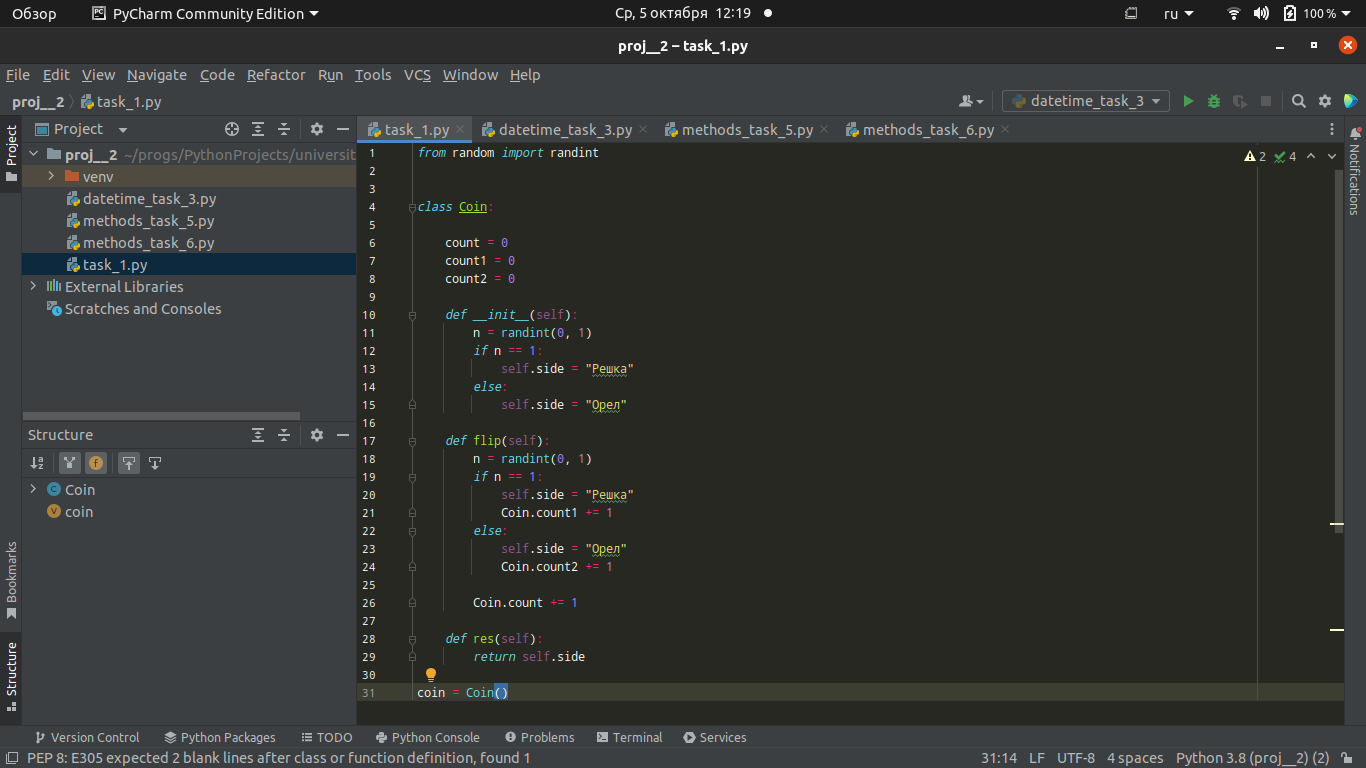


Рис 1.1 – создание класса Coin

В переменную count будем записывать кол-во подбрасываний, а в count1 и count2, кол-во выпадений каждой из сторон. В конструкторе, с помощью функции randint модуля random, случайно будем устанавливать начальную сторону — side (Рис 1.2)

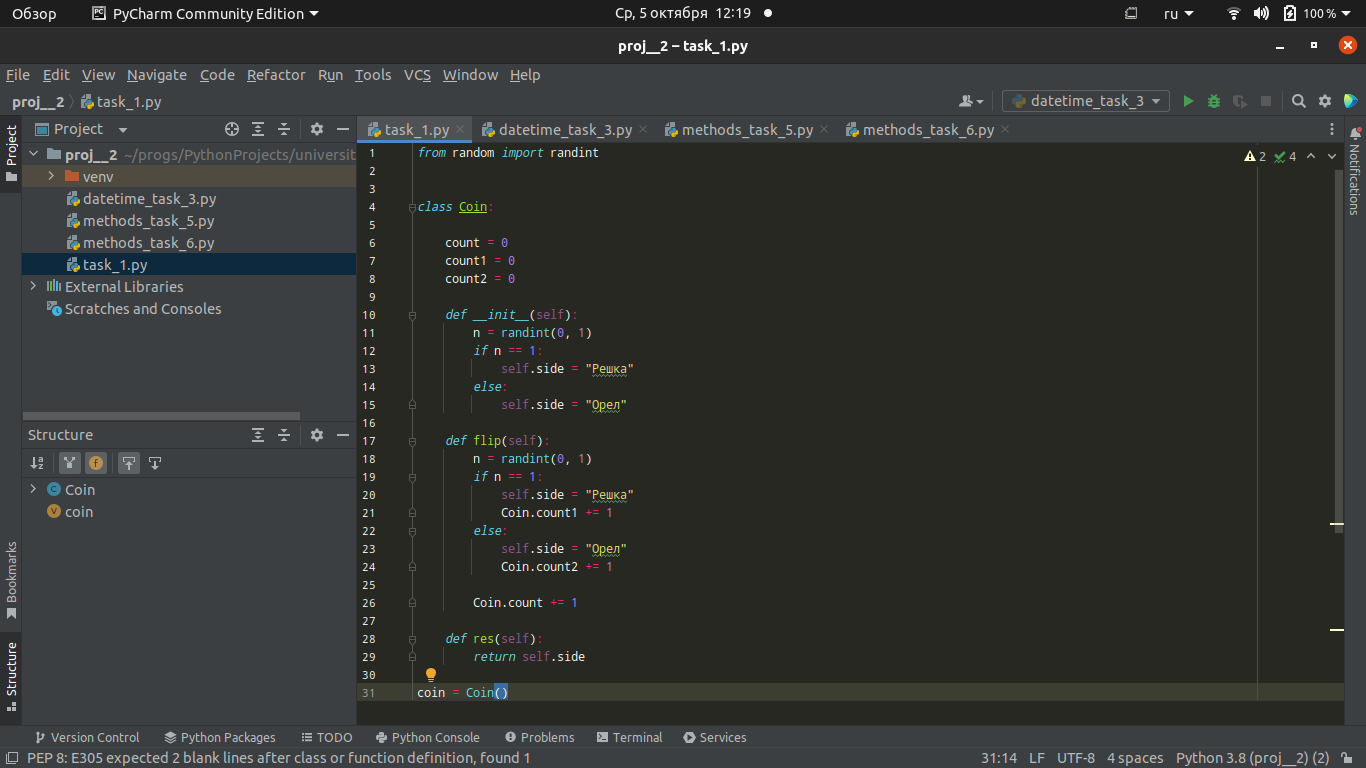


Рис 1.2 – конструктор класса Coin

Используя ту же функцию randint, будем имитировать подбрасывание монеты в функции flip и каждый раз увеличивать count на 1 (Рис 1.3).

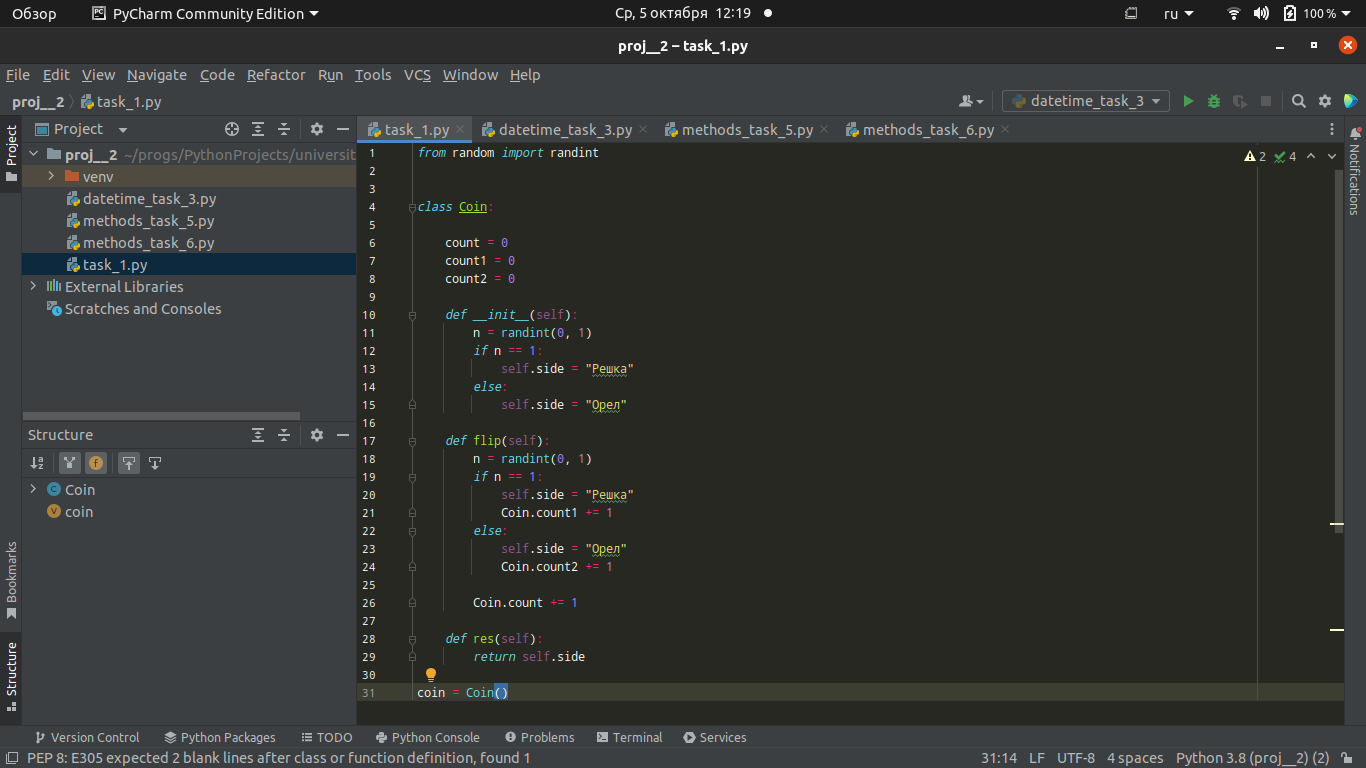


Рис 1.3 – функция flip класса Coin

Функция res будет возвращать текущую сторону монеты (Рис 1.4).

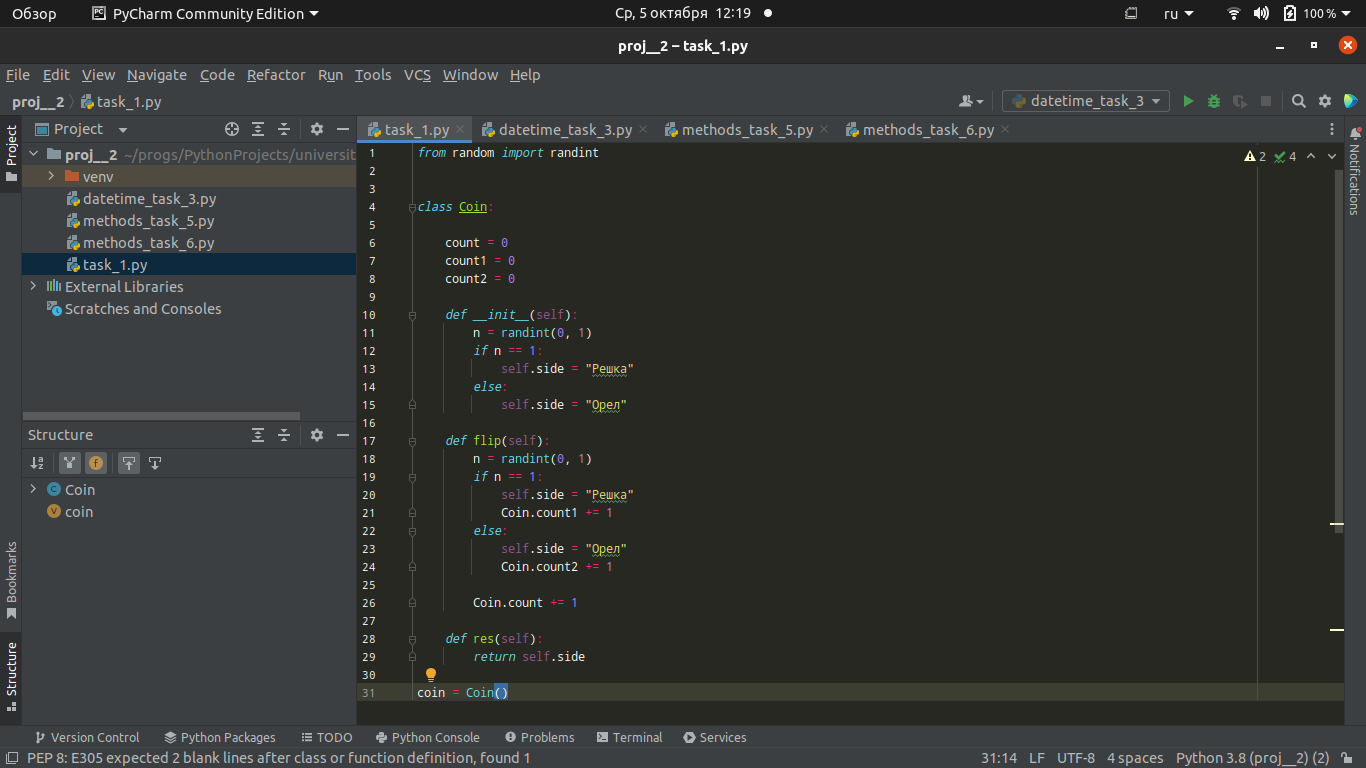


Рис 1.4 – функция res класса Coin

Создадим объект монеты coin. Запустим цикл, чтобы подбросит монету 10 раз (Рис 1.5).

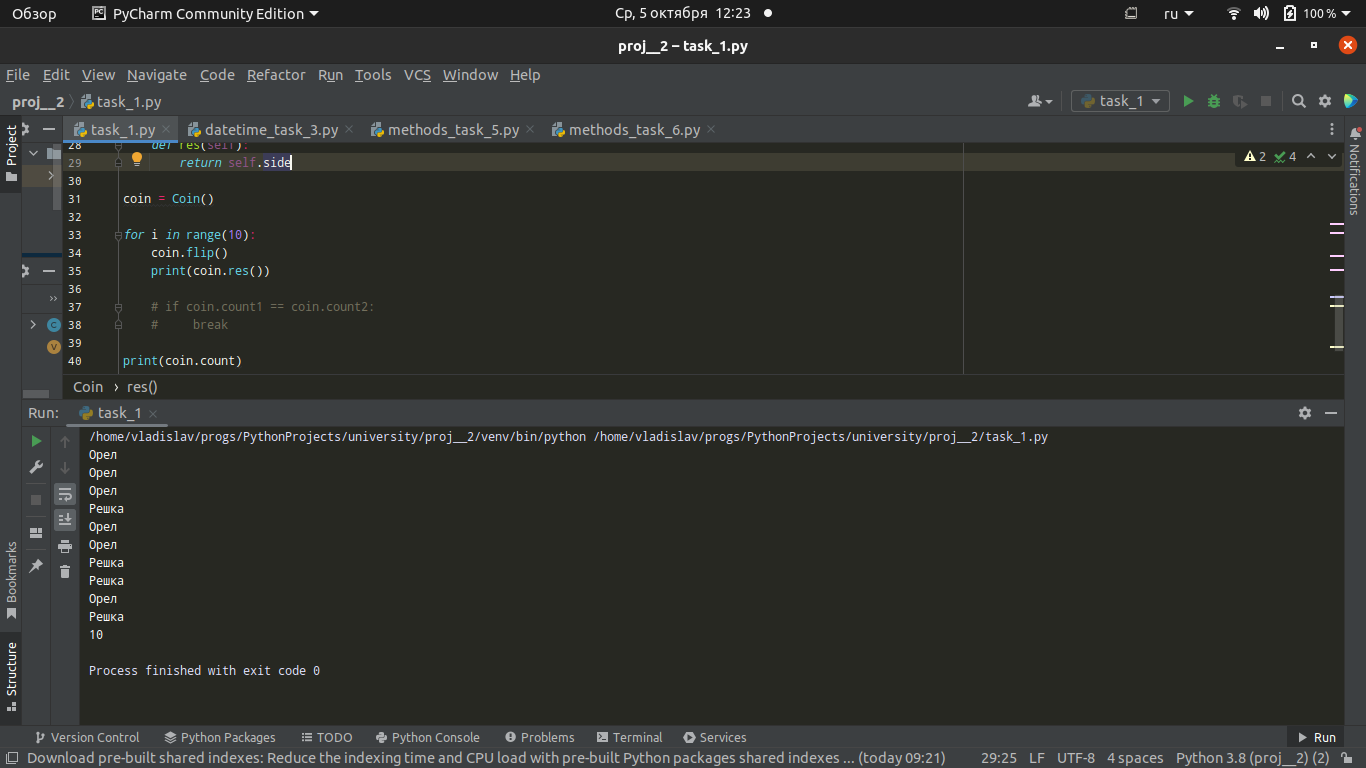


Рис 1.5 – результаты подбрасывания монеты

Определим равновероятность с помощью сравнения кол-ва выпадений каждой из сторон. Если они равны, цикл заканчивается (Рис 1.6).

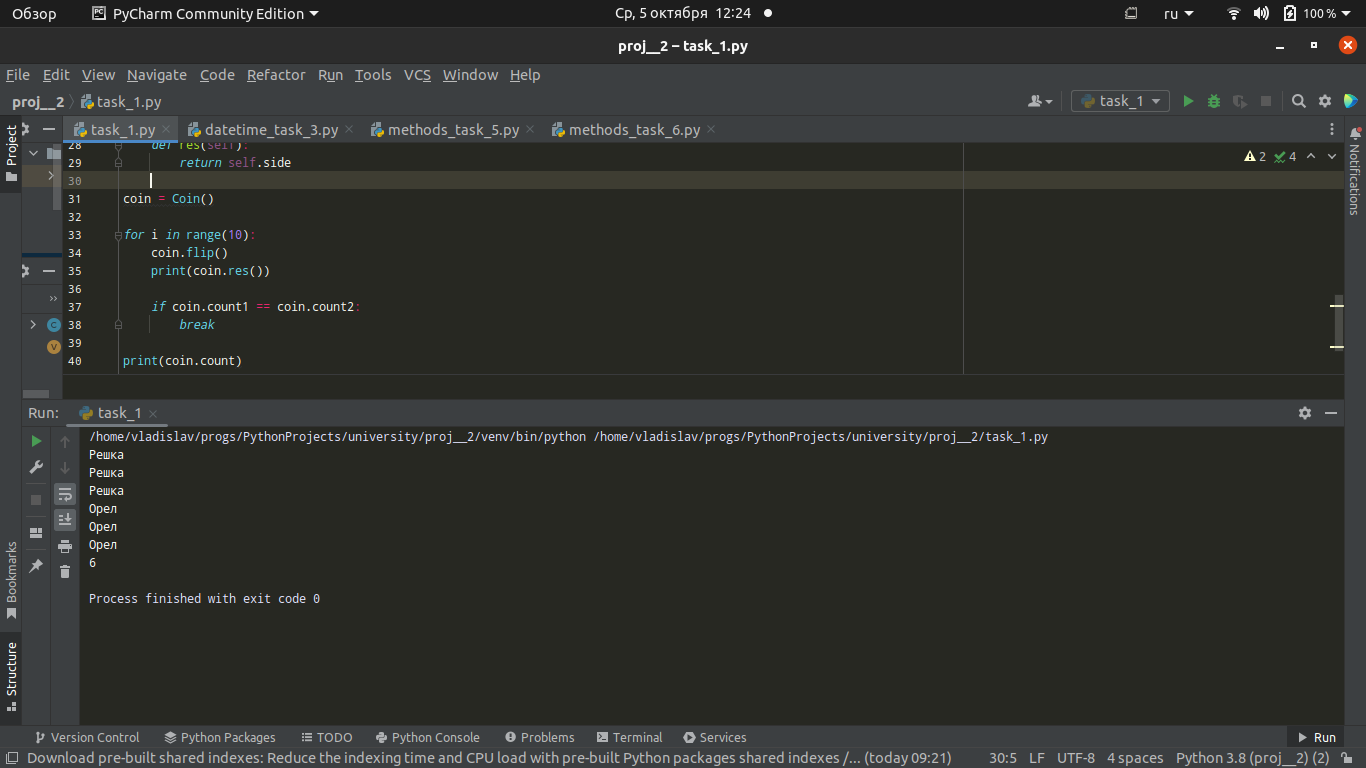


Рис 1.6 – результаты подбрасывания монеты с определением равновероятности

Теперь создадим три объекта монеты и каждый подбросим 5 раз. Посмотрим на результат (Рис 1.7).

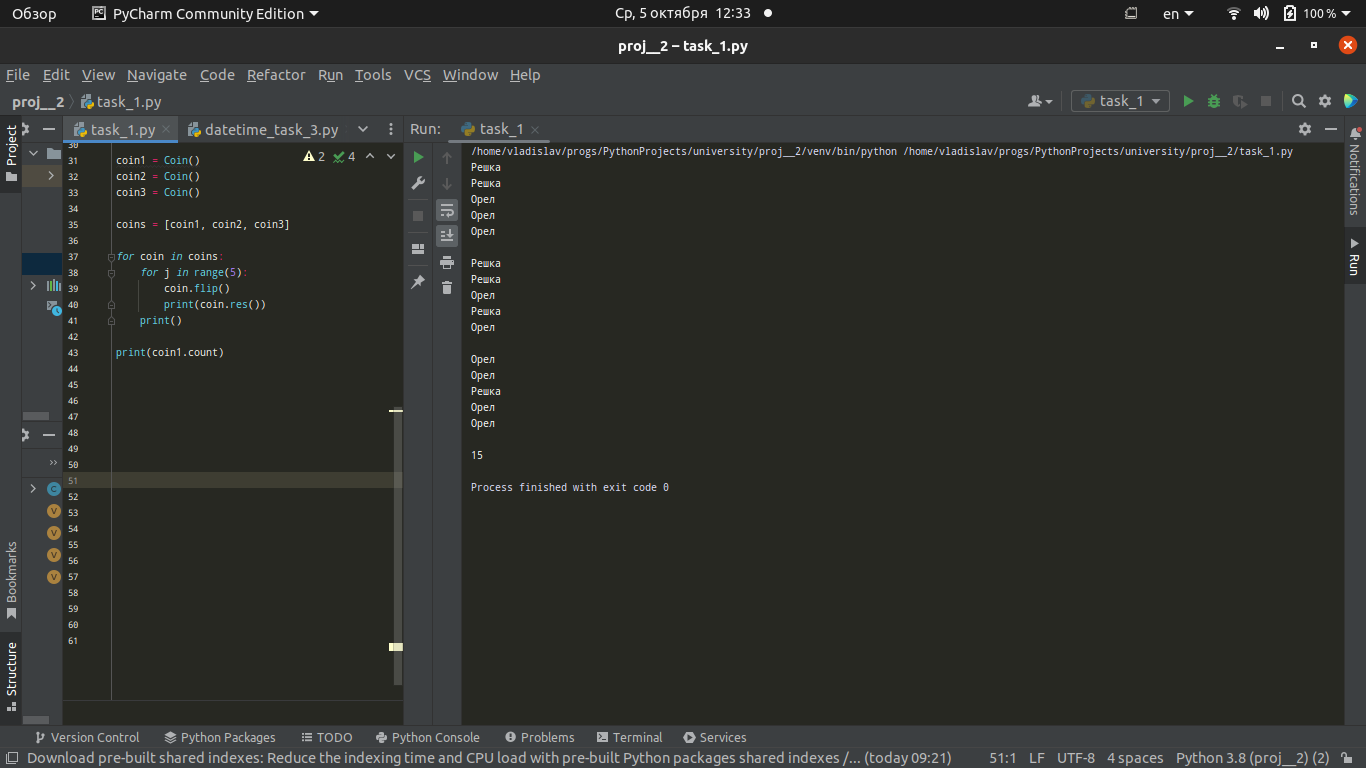


Рис 1.7 – результаты подбрасывания монет

**Итог**

Мы создали класс-имитатор подбрасывания монеты. Провели тесты с подбрасываниями объектов монет и равновероятностью исходов.

**Задание 3**

**Текст задания**

Ознакомиться с особенностями работы модуля datetime. Реализовать несколько примеров.

**Примеры работы модуля datetime (Рис 2.1.1-2.1.2)**

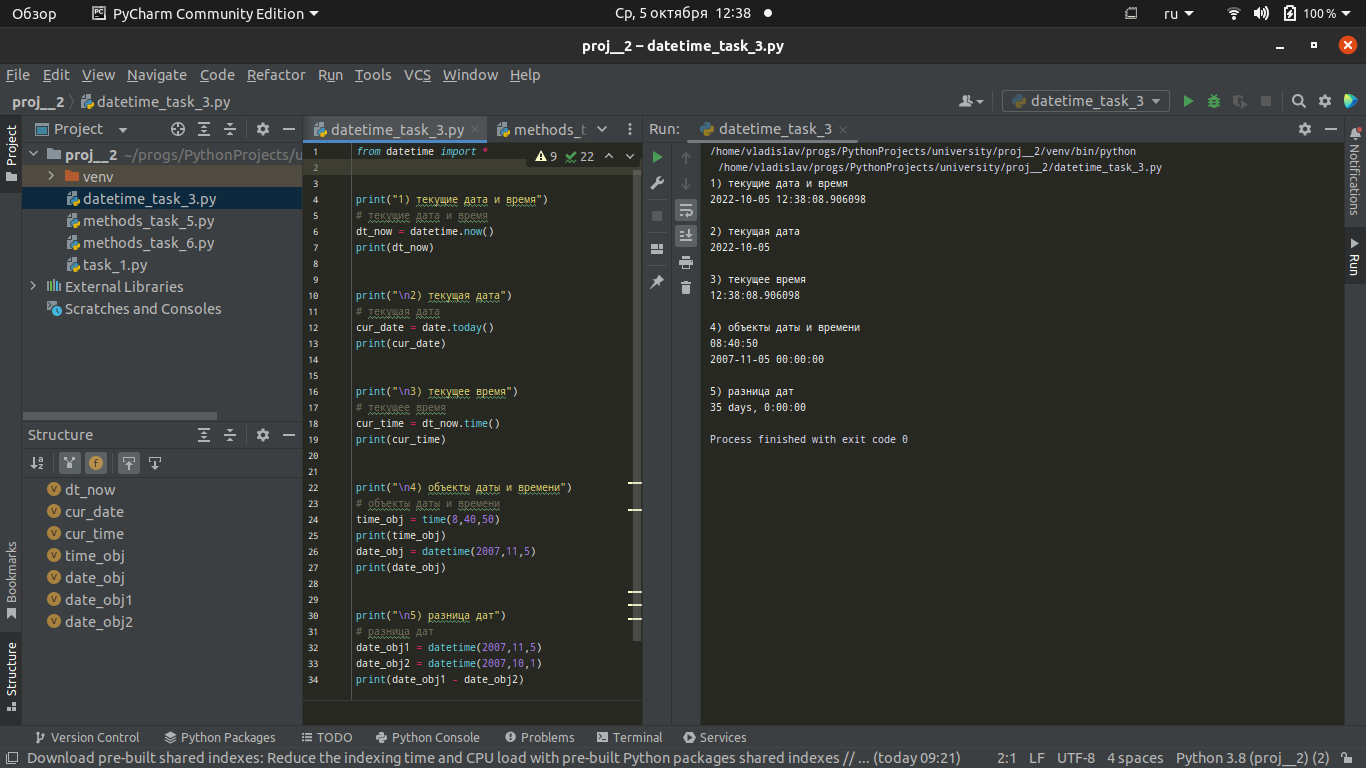


Рис 2.1.1 - скрипт программы

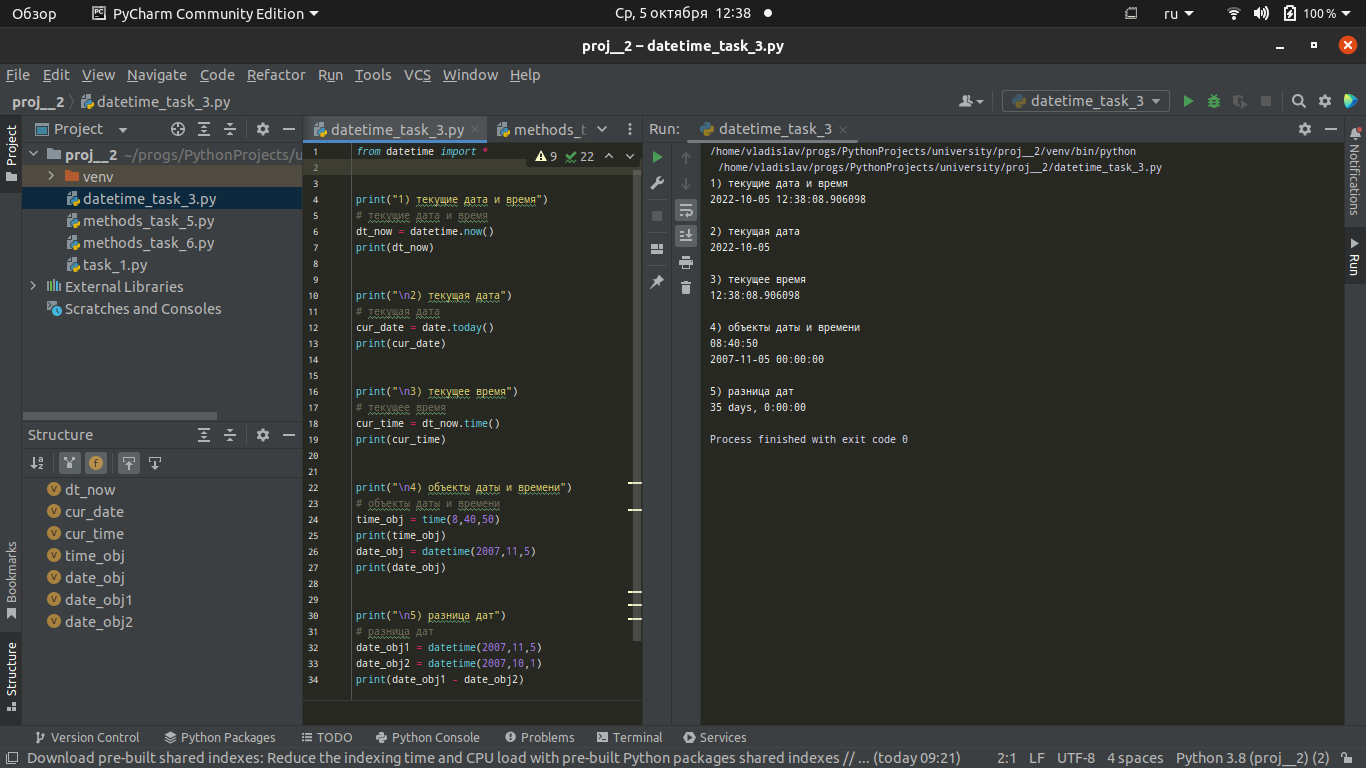


Рис 2.1.2 – вывод программы

**Задание 5**

**Текст задания**

Изучить особенности работы методов str, repr, \_\_str\_\_ и \_\_repr\_\_ , реализовав в PyCharm примеры соответствующих рисунков в лекции.

**Примеры работы методов str, repr, \_\_str\_\_ и \_\_repr\_\_ (Рис 3.1.1-3.1.3)**

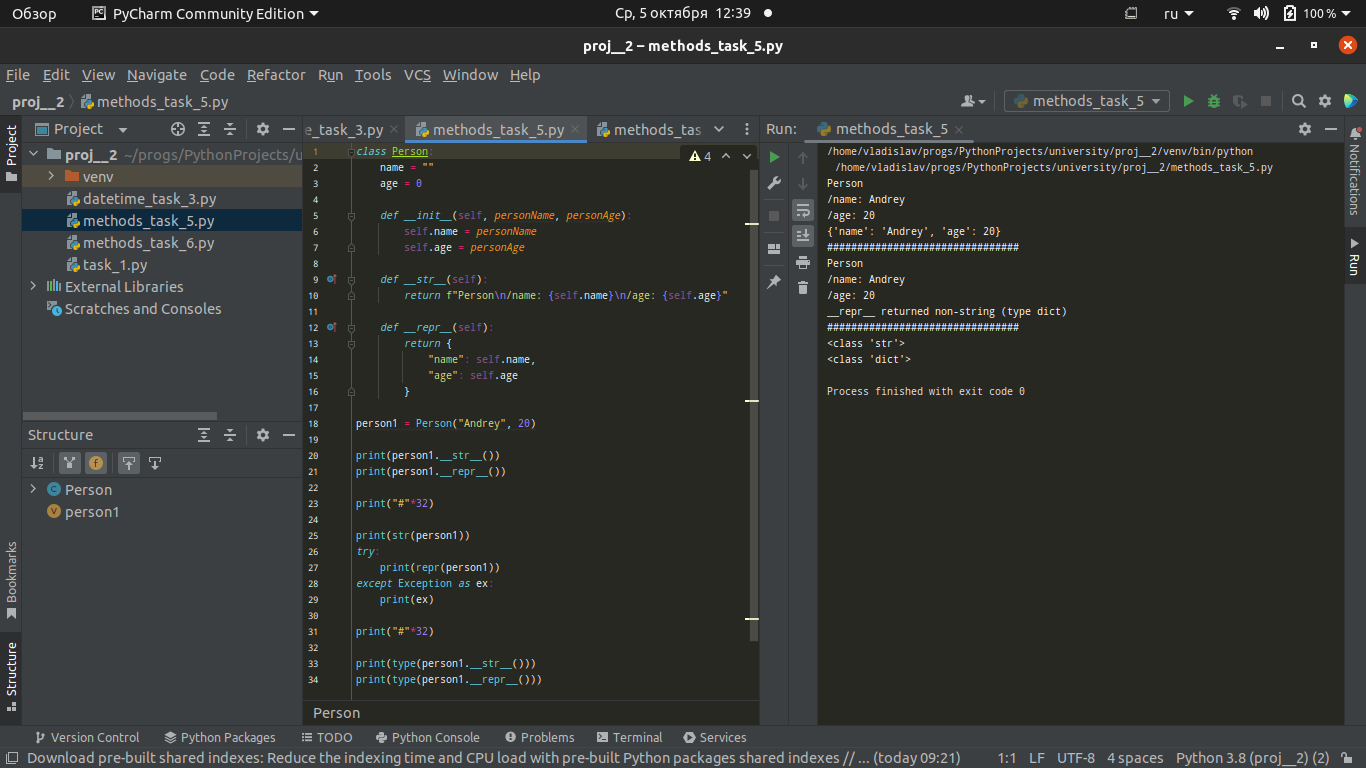


Рис 3.1.1 – класс Person

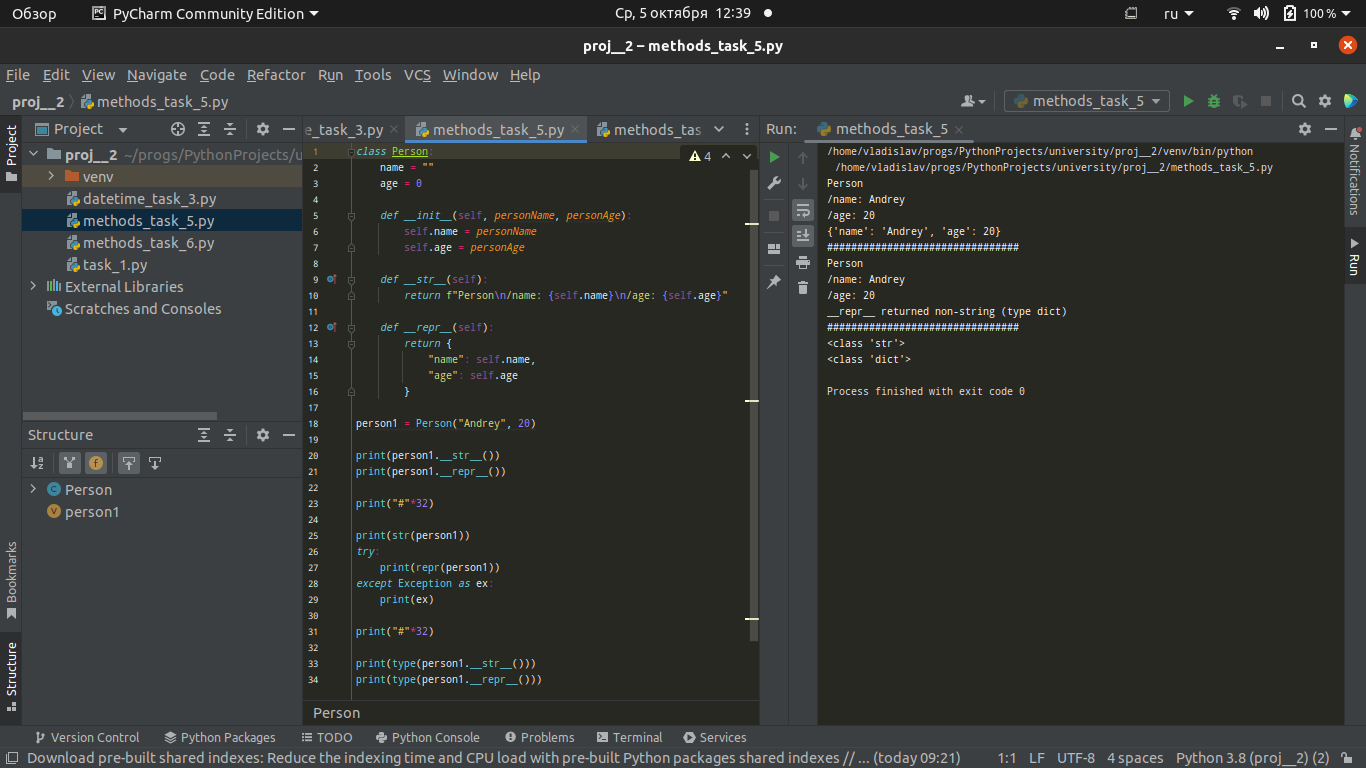


Рис 3.1.2 - скрипт программы

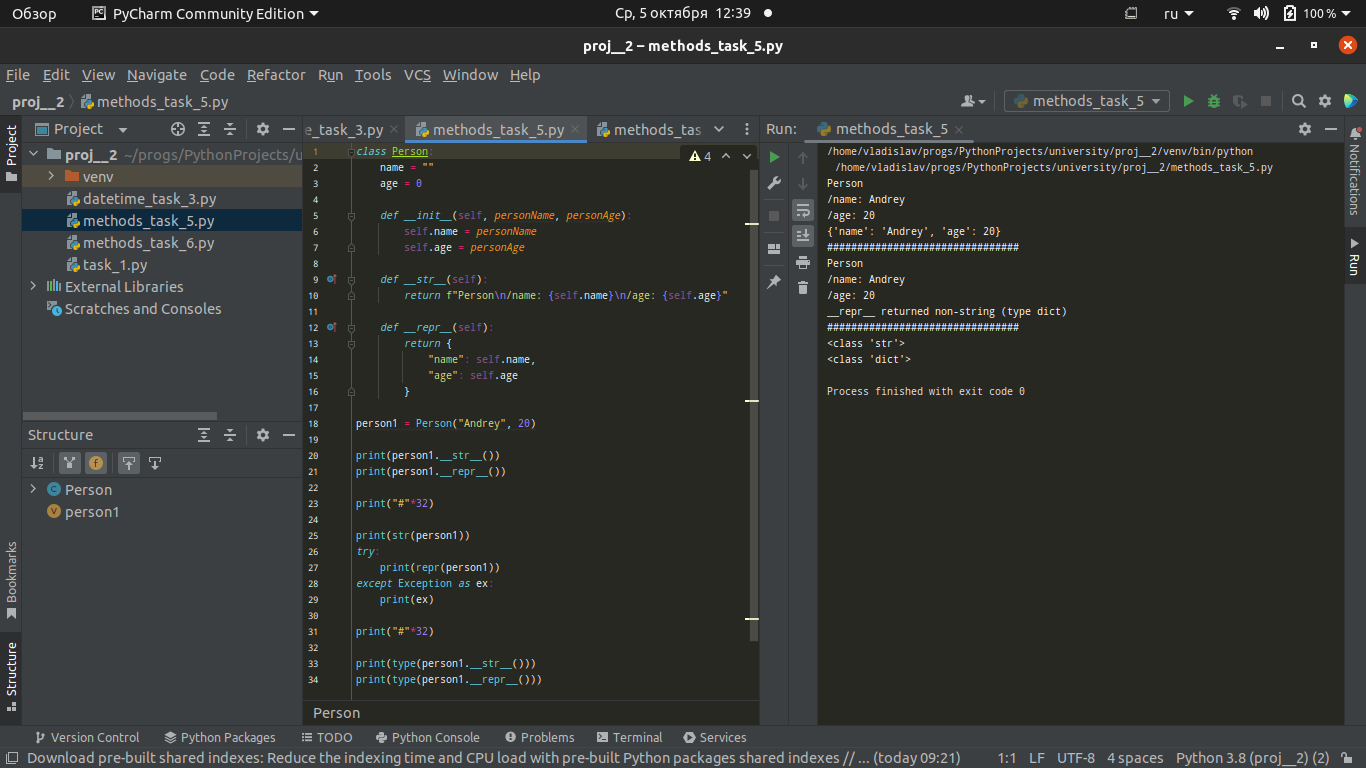


Рис 3.1.3 - результат работы программы

**Задание 6**

**Текст задания**

Смоделировать работу магических методов на примере класса «Курс».

**Работа магических методов \_\_len\_\_, \_\_getitem\_\_, \_\_setitem\_\_ (Рис 4.1.1-4.1.3)**

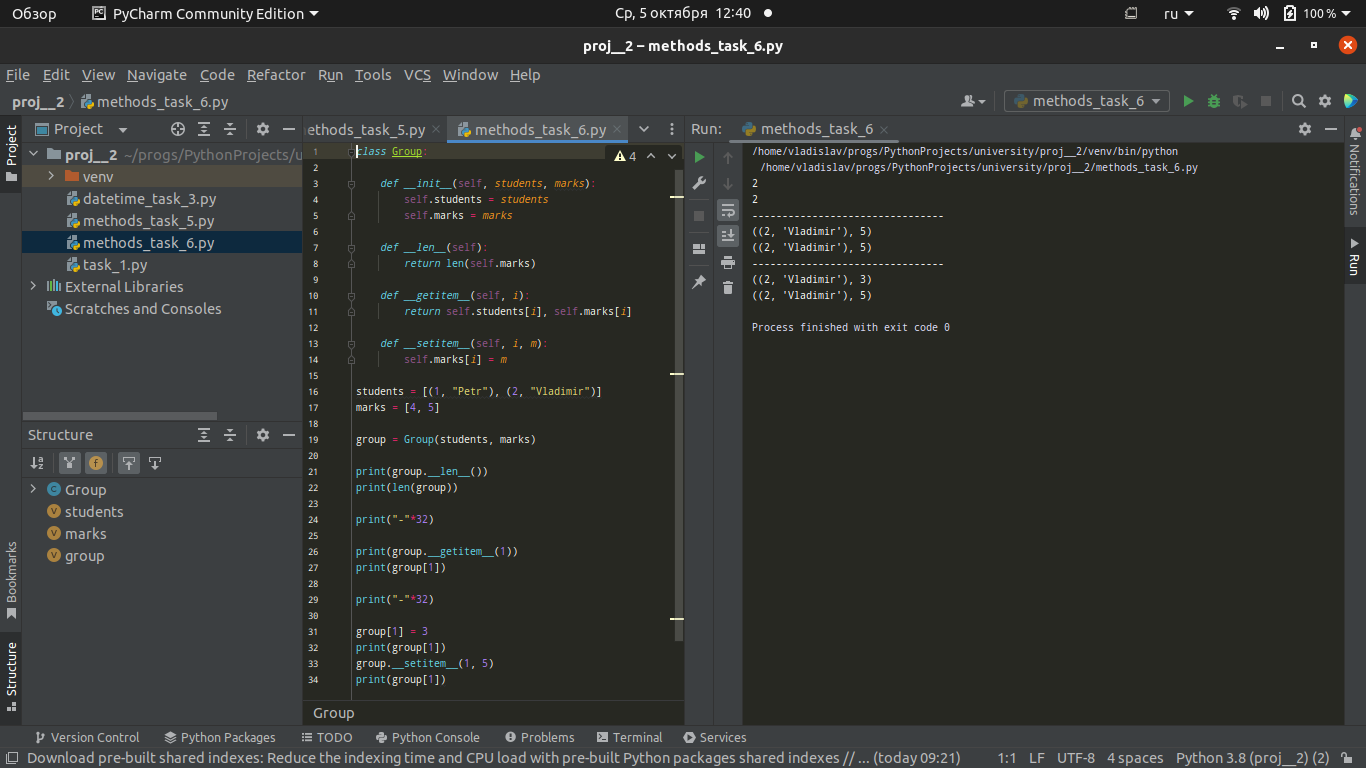


Рис 4.1.1 – класс Group

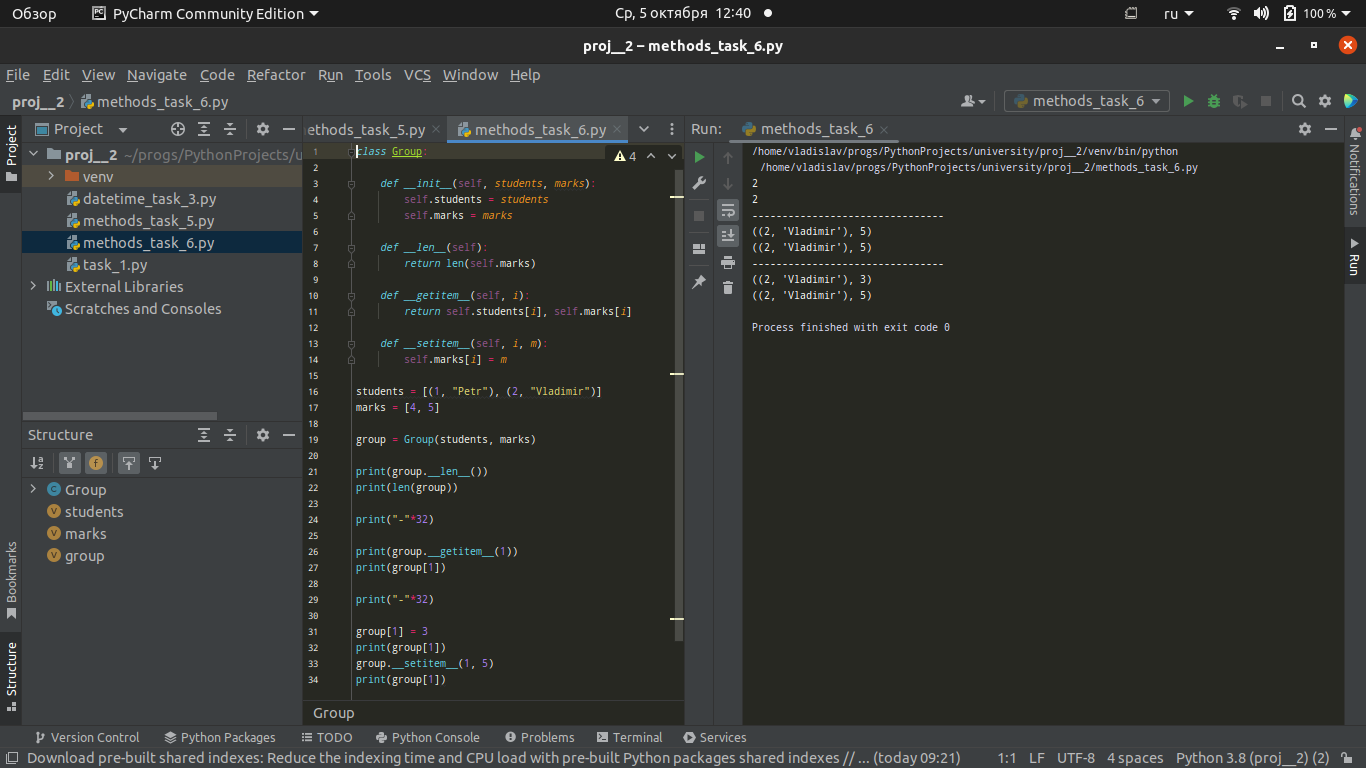


Рис 4.1.2 - скрипт программы

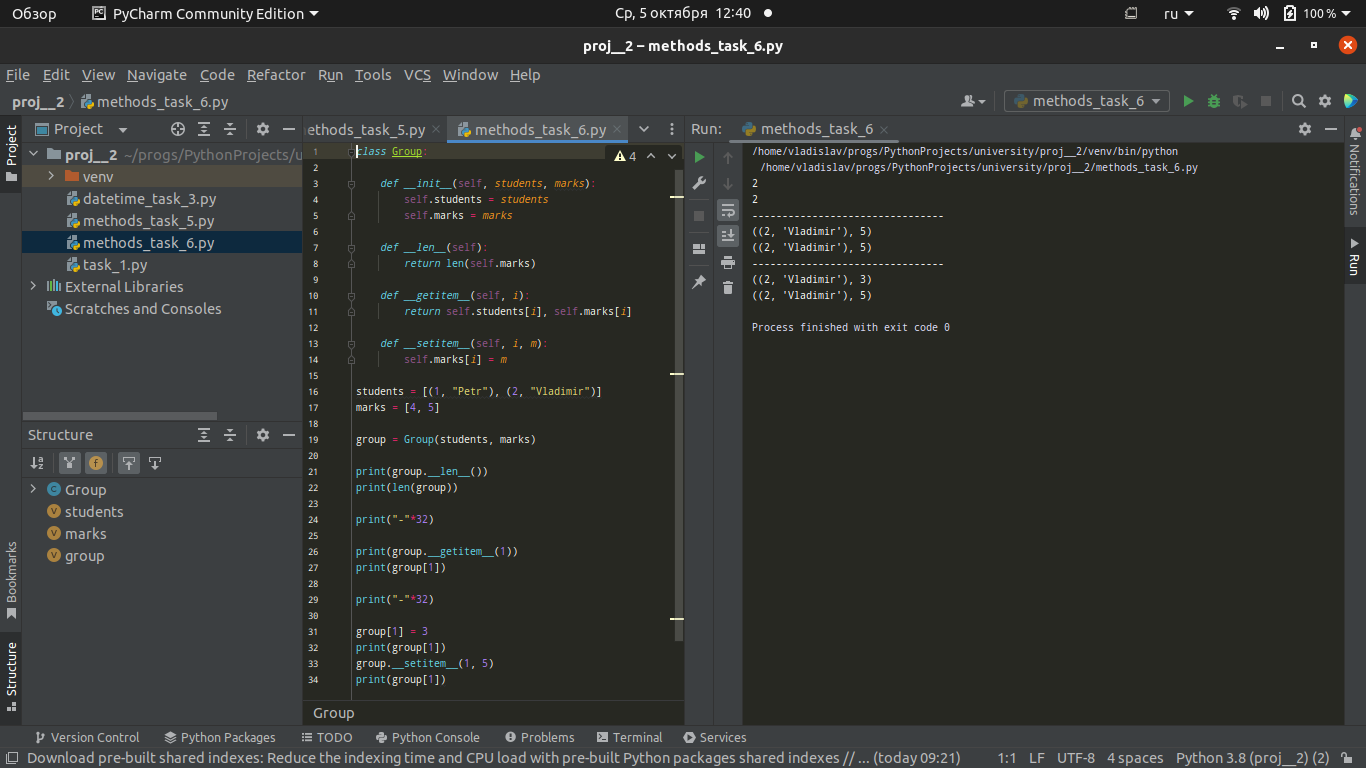


Рис 4.1.3 - результат работы программы