МИНИCTEPCTBO НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное

образовательное учреждение высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

**Отчет по лабораторной работе №4.1**

**Элементы объектно-ориентированного программирования в языке Python**

**По дисциплине «Объектно-ориентированное программирование»**

Выполнил студент группы ИВТ-б-о-20-1

Плотников Д. В. « »\_\_\_\_\_\_\_\_\_20\_\_г.

Подпись студента\_\_\_\_\_\_\_\_\_\_\_

Работа защищена « »\_\_\_\_\_\_\_\_\_\_\_\_20\_\_г.

Проверил Воронкин Р. А. \_\_\_\_\_\_\_\_

(подпись)

Ставрополь 2022

**Цель работы:** приобретение навыков по работе с классами и объектами при написании программ с помощью языка программирования Python версии 3.x.

**Ход работы**

1. Создал новый собственный репозиторий. Ссылка на репозиторий: <https://github.com/Dmitry-15/4.1_laba>.
2. С помощью команды git clone клонировал удаленный репозиторий на свой ПК. Дополнил файл .gitignore необходимыми правилами для работы с IDE PyCharm.
3. Далее проработал пример.

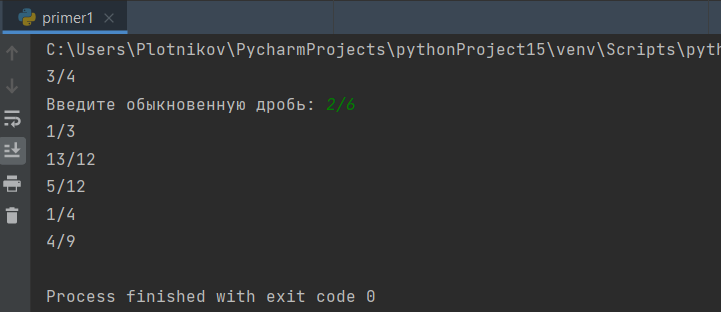


Рисунок 1 – Выполнение примера

**Индивидуальные задания**

**Вариант 14**

**Задание 1**

1. Условие задачи: поле first — дробное положительное число, оклад; поле second — целое число, количество отработанных дней в месяце. Реализовать метод summa () — вычисление начисленной суммы за данное количество дней для заданного месяца: оклад / дни месяца \* отработанные дни.

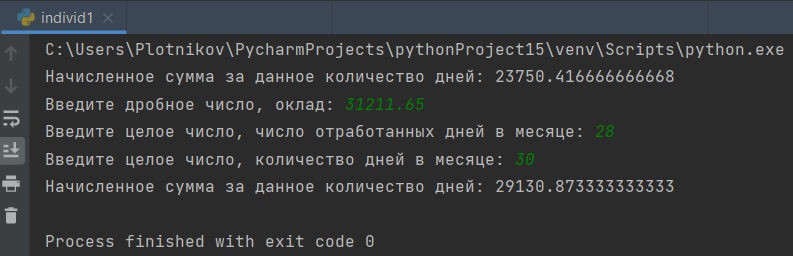


Рисунок 2 – Выполнение первого индивид. задания

**Задание 2**

1. Условие задачи: создать класс Payment (зарплата). В классе должны быть представлены поля: фамилия-имя-отчество, оклад, год поступления на работу, процент надбавки, подоходный налог, количество отработанных дней в месяце, количество рабочих дней в месяце, начисленная и удержанная суммы. Реализовать методы: вычисления начисленной суммы, вычисления удержанной суммы, вычисления суммы, выдаваемой на руки, вычисления стажа. Стаж вычисляется как полное количество лет, прошедших от года поступления на работу, до текущего года. Начисления представляют собой сумму, начисленную за отработанные дни, и надбавки, то есть доли от первой суммы. Удержания представляют собой отчисления в пенсионный фонд (1% от начисленной суммы) и подоходный налог. Подоходный налог составляет 13% от начисленной суммы без отчислений в пенсионный фонд.

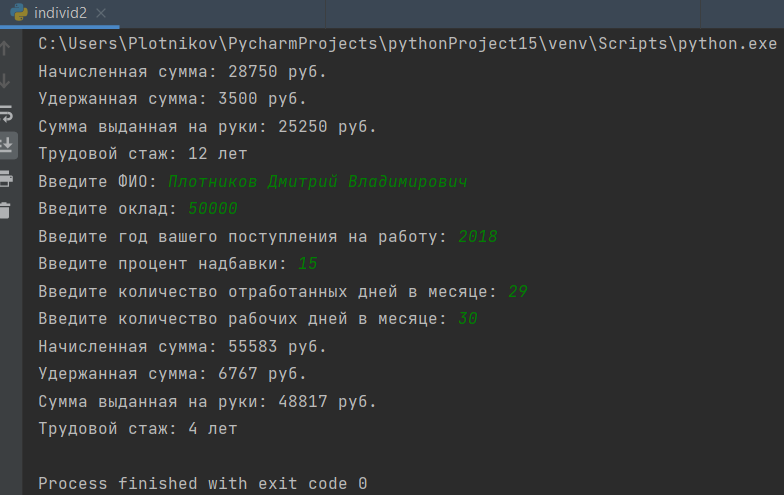


Рисунок 3 – Выполнение второго индивид. задания

# Контрольные вопросы:

# Как осуществляется объявление класса в языке Python?

Классы объявляются с помощью ключевого слова class и имени класса.

1. Чем атрибуты класса отличаются от атрибутов экземпляра?

Атрибуты класса определены внутри класса, но вне каких-либо методов. Их значения одинаковы для всех экземпляров этого класса. Так что вы можете рассматривать их как тип значений по умолчанию для всех наших объектов. Что касается переменных экземпляра, они хранят данные, уникальные для каждого объекта класса.

1. Каково назначение методов класса?

Методы определяют функциональность объектов, принадлежащих конкретному классу.

1. Для чего предназначен метод \_\_init\_\_() класса?

Метод \_\_init\_\_ является конструктором. Конструкторы - это концепция объектно-ориентированного программирования. Класс может иметь один и только один конструктор. Если \_\_init\_\_ определен внутри класса, он автоматически вызывается при создании нового экземпляра класса.

1. Каково назначение self?

Аргумент self представляет конкретный экземпляр класса и позволяет нам получить доступ к его атрибутам и методам. В примере с \_\_init\_\_ мы создаем атрибуты для конкретного экземпляра и присваиваем им значения аргументов метода. Важно использовать параметр self внутри метода, если мы хотим сохранить значения экземпляра для последующего использования.

1. Как добавить атрибуты в класс?

Атрибут класса – это атрибут, общий для всех экземпляров класса. Атрибуты экземпляра – это как раз те, которые мы определяем в методах, поэтому по определению мы можем создавать новые атрибуты внутри наших пользовательских методов.

1. Как осуществляется управление доступом к методам и атрибутам вязыке Python?

В Python таких возможностей нет, и любой может обратиться к атрибутам и методам вашего класса, если возникнет такая необходимость. Это существенный недостаток этого языка, т.к. нарушается один из ключевых принципов ООП – инкапсуляция. Хорошим тоном считается, что для чтения/изменения какого-то атрибута должны использоваться специальные методы, которые называются getter/setter, их можно реализовать, но ничего не помешает изменить атрибут напрямую. При этом есть соглашение, что метод или атрибут, который начинается с нижнего подчеркивания, является скрытым, и снаружи класса трогать его не нужно (хотя сделать это можно).

1. Каково назначение функции isinstance?

В Python есть встроенная функция instance (), которая сравнивает значение с указанным типом. Если данное значение и тип соответствуют, он вернет true, иначе false. Используя isinstance (), вы можете проверить строку, число с плавающей точкой, int, список, кортеж, dict, set, class и т.д.

**Вывод:** в ходе выполнения лабораторной работы приобрел навыки по работе с классами и объектами при написании программ с помощью языка программирования Python версии 3.x.