

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт перспективной инженерии
Департамент цифровых, робототехнических систем и электроники

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №3
дисциплины
«Объектно-ориентированное программирование»

Выполнил:
Евдаков Евгений Владимирович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

Проверил:
Воронкин Р. А., доцент департамента
цифровых, робототехнических систем
и электроники института
перспективной инженерии

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 г.

Тема: наследование и полиморфизм в языке Python.

Цель: приобретение навыков по созданию иерархии классов при написании программ с помощью языка программирования Python версии 3.x.

Порядок выполнения работы:

Задание 1. Создал общедоступный репозиторий на GitHub, в котором использована лицензий MIT и язык программирования Python, также добавил файл .gitignore с необходимыми правилами. Клонировал свой репозиторий на свой компьютер. Организовал свой репозиторий в соответствие с моделью ветвления git-flow, появилась новая ветка develop в которой буду выполнять дальнейшие задачи.

```
C:\Users\evdak>git clone https://github.com/EvgenyEvdakov/Laba_4.3.git
Cloning into 'Laba_4.3'...
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 21 (delta 1), reused 16 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (21/21), 8.77 KiB | 998.00 KiB/s, done.
Resolving deltas: 100% (1/1), done.
```

Рисунок 1. Клонирование репозитория

Задание 2. Создал виртуальное окружение conda и активировал его, также установил необходимые пакеты isort, black, flake8.

```
(base) PS C:\Users\evdak> cd C:\Users\evdak\Laba_4.3
(base) PS C:\Users\evdak\Laba_4.3> conda create -n oop_1 python=3.10
Retrieving notices: ...working... done
Channels:
 - defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\evdak\.conda\envs\oop_1

  added / updated specs:
    - python=3.10

The following NEW packages will be INSTALLED:

  bzip2                pkgs/main/win-64::bzip2-1.0.8-h2bfff1b_6
  ca-certificates      pkgs/main/win-64::ca-certificates-2024.9.24-haa95532_0
  libffi               pkgs/main/win-64::libffi-3.4.4-hd77b12b_1
  openssl              pkgs/main/win-64::openssl-3.0.15-h827c3e9_0
  pip                  pkgs/main/win-64::pip-24.2-py310haa95532_0
```

Рисунок 2. Создание виртуального окружения

Задание 3. Создал проект PyCharm в папке репозитория. Приступил к работе с примером. Добавил новый файл primer1.py.

Условие примера: Рациональная (несократимая) дробь представляется парой целых чисел (a, b), где a — числитель, b —

знаменатель. Создать класс Rational для работы с рациональными дробями. Обязательно должны быть реализованы операции:

- сложения add, $(a, b) + (c, d) = (ad + be, bd)$;
- вычитания sub, $(a, b) - (c, d) = (ad - be, bd)$;
- умножения mul, $(a, b) \times (c, d) = (ac, bd)$;
- деления div, $(a, b) / (c, d) = (ad, be)$;
- сравнения equal, greater, less.

Должна быть реализована приватная функция сокращения дроби reduce, которая обязательно вызывается при выполнении арифметических операций. Напишем программу для решения поставленной задачи:

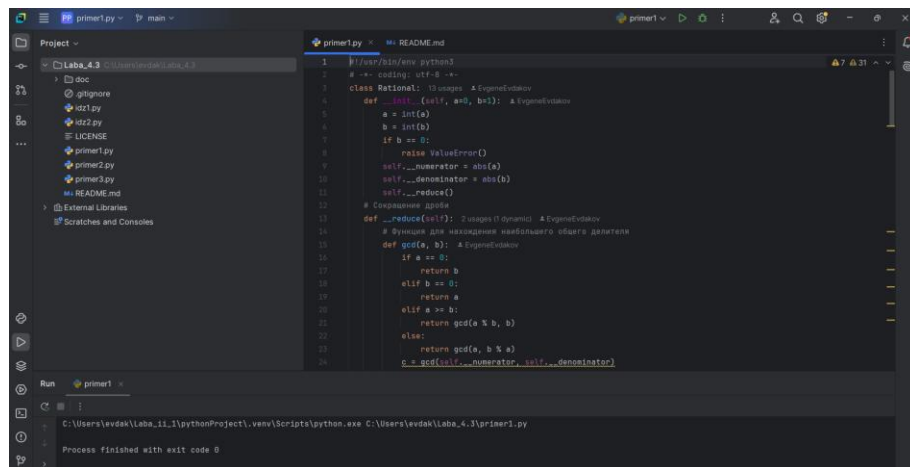


Рисунок 3. Выполнение первого примера

Далее приступил к работе со вторым примером. Добавил новый файл primer2.py.

Условие примера: во втором примере необходимо изучить, как работают абстрактные классы в Python.

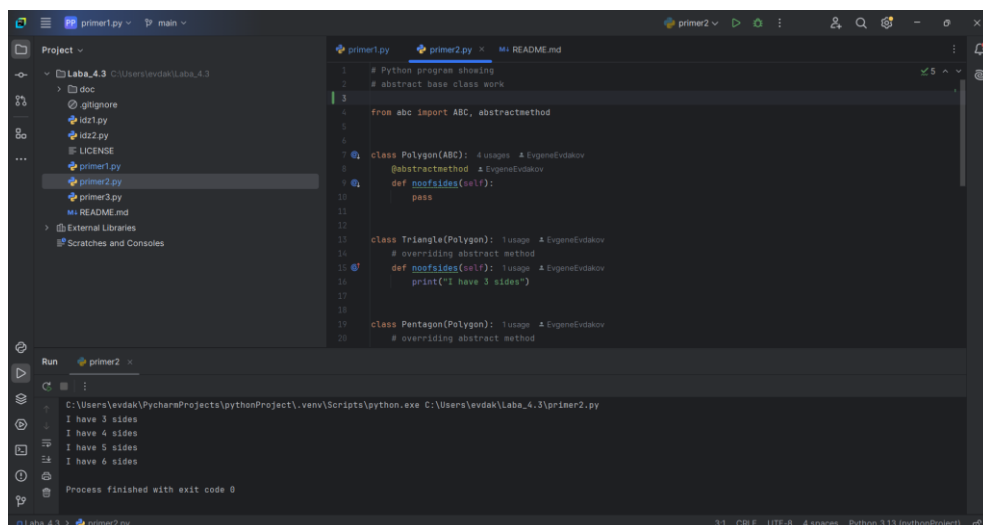


Рисунок 4. Выполнение второго примера

Далее приступил к работе с третьим примером. Добавил новый файл primer3.py.

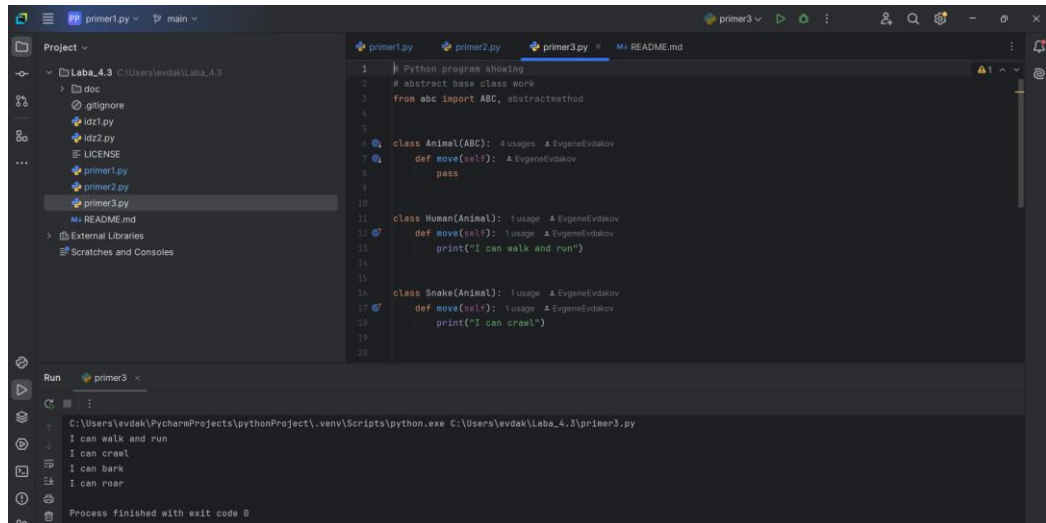


Рисунок 5. Выполнение третьего примера

Задание 4.

Индивидуальное задание

Вариант 6

Создал новый файл под названием idz1.py.

Условие задания: необходимо создать класс Triad (тройка чисел); определить методы изменения полей и вычисления суммы чисел. Определить производный класс Triangle с полями-сторонами. Определить методы вычисления углов и площади треугольника.

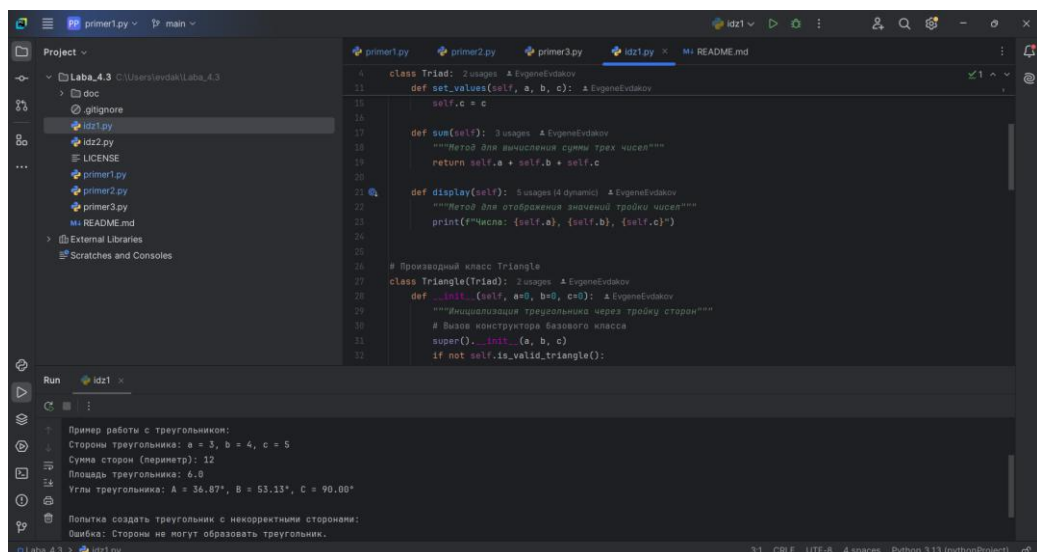


Рисунок 6. Выполнение первого индивидуального задания

Создал новый файл под названием idz2.py.

Условие задания: необходимо реализовать абстрактный базовый класс, определив в нем абстрактные методы и свойства. Эти методы определяются в производных классах. В базовых классах должны быть объявлены абстрактные методы ввода/вывода, которые реализуются в производных классах.

Вызывающая программа должна продемонстрировать все варианты вызова переопределенных абстрактных методов. Написать функцию вывода, получающую параметры базового класса по ссылке и демонстрирующую виртуальный вызов.

Создать абстрактный базовый класс `Pair` с виртуальными арифметическими операциями. Создать производные классы `FazzyNumber` (нечеткое число) и `Complex` (комплексное число).

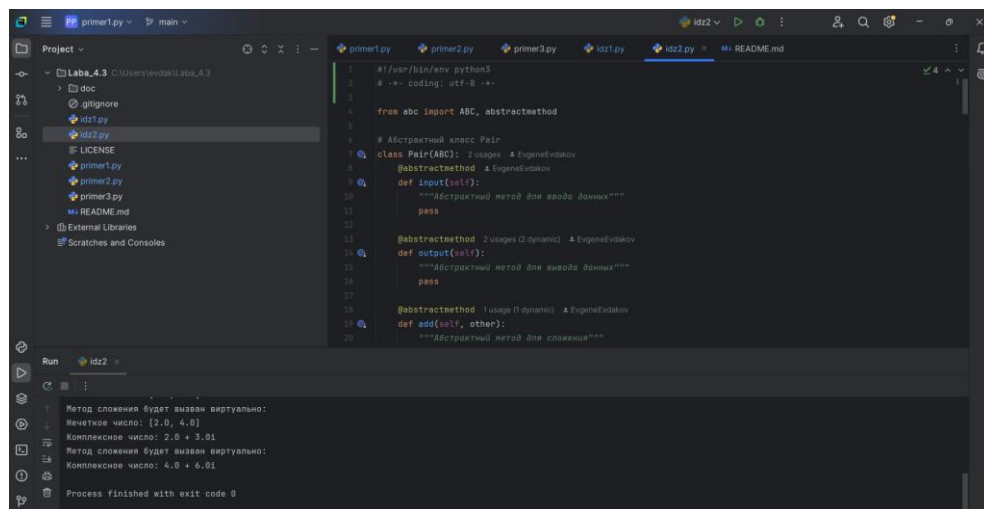


Рисунок 7. Выполнение второго индивидуального задания

Задание 5.

После выполнения работы на ветке `develop`, слил ее с веткой `main` и отправил изменения на удаленный сервер. Создал файл `envirement.yml` и деактивировал виртуальное окружение.

```
(oop_1) PS C:\Users\evdak\Laba_4.3> conda export > environment
(oop_1) PS C:\Users\evdak\Laba_4.3> conda deactivate
```

Рисунок 8. Деактивация VO

Ссылка: https://github.com/EvgenyEvdakov/Laba_4.3

Ответы на контрольные вопросы:

1. Что такое наследование как оно реализовано в языке Python?

Наследование — это принцип объектно-ориентированного программирования, который позволяет создать новый класс (производный, или дочерний) на основе существующего класса (базового, или родительского). Производный класс наследует методы и свойства базового класса, что позволяет избежать дублирования кода. В Python наследование реализуется указанием базового класса в скобках при объявлении нового класса.

2. Что такое полиморфизм и как он реализован в языке Python?

Полиморфизм — это способность объекта обрабатывать методы с одинаковыми именами, но с различным поведением, в зависимости от типа данных или класса. В Python полиморфизм реализован через переопределение методов: дочерние классы могут переопределять методы, унаследованные от родительских классов, для создания специализированного поведения.

3. Что такое "утиная" типизация в языке программирования Python?

"Утиная" типизация (duck typing) — это концепция, при которой в языке программирования проверяется не тип объекта, а его поведение. Если объект имеет нужные методы или свойства, он может использоваться в данном контексте, независимо от его конкретного типа. В Python "утиная" типизация позволяет применять объекты любого класса, если они поддерживают необходимые методы.

4. Каково назначение модуля abc языка программирования Python?

Модуль abc (Abstract Base Classes) предоставляет средства для определения абстрактных базовых классов и создания абстрактных методов. Абстрактный класс — это класс, который нельзя создать напрямую; он служит как основа для других классов. Модуль abc позволяет реализовывать

абстракции в Python, обеспечивая, чтобы дочерние классы реализовали обязательные методы.

5. Как сделать некоторый метод класса абстрактным?

Чтобы сделать метод класса абстрактным, нужно импортировать ABC и abstractmethod из модуля abc и аннотировать метод с помощью декоратора @abstractmethod. Абстрактный метод — это метод, который обязан быть реализован в дочерних классах. Класс, содержащий абстрактные методы, не может быть создан напрямую.

6. Как сделать некоторое свойство класса абстрактным?

Чтобы сделать свойство класса абстрактным, в модуле abc используется декоратор @property совместно с @abstractmethod. Это позволяет заставить дочерние классы реализовывать конкретное свойство.

7. Каково назначение функции isinstance?

Функция isinstance проверяет, является ли объект экземпляром указанного класса или кортежа классов. Она возвращает True, если объект принадлежит указанному классу (или одному из классов в кортеже), и False в противном случае. Эта функция полезна для проверки типа данных, особенно в полиморфных структурах.

Вывод: приобрел навыки по созданию иерархии классов при написании программ с помощью языка программирования Python версии 3.x.