

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

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №4.3
«Наследованиеи полиморфизм в языке Python»

Выполнила:
Кузнецова Алена Валерьевна
3 курс, группа ИВТ-б-о-21-1,
09.03.01 «Информатика
и вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем»,
очная форма обучения

(подпись)

Ставрополь, 2023 г.

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

Выполнение работы:

1. Создали общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python.

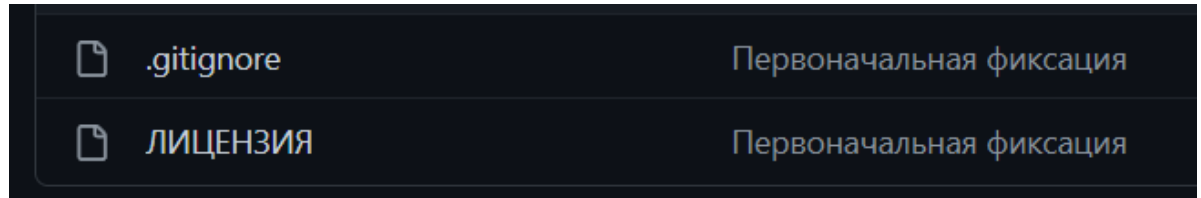


Рисунок 1 – Создание репозитория

2. Выполнили клонирование созданного репозитория.

```
C:\Users\akuzn>cd /d C:\учеба\2 курс\3 семестр\Программирование Python\Репозитории
C:\учеба\2 курс\3 семестр\Программирование Python\Репозитории>git clone https://github.com/AlenaKuzn/2.6.git
Cloning into '2.6'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
```

Рисунок 2 – Клонирование репозитория

3. Организовали свой репозиторий в соответствии с моделью git-flow.

```
C:\учеба\2 курс\3 семестр\Программирование Python\Репозитории\2.6>git flow init
Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/учеба/2 курс/3 семестр/Программирование Python/Репозитории/2.6/.git/hooks]
```

Рисунок 3 – Моделью ветвления git-flow

4. Выполнение индивидуального задания №1.

Составить программу с использованием иерархии классов. Номер варианта необходимо получить у преподавателя. В раздел программы, начинающийся после инструкции `if __name__ == '__main__':` добавить код, демонстрирующий возможности разработанных классов. Создать класс `Pair` (пара чисел); определить методы изменения полей и вычисления

произведения чисел. Определить производный класс RightAngled с полями-катетами. Определить методы вычисления гипотенузы и площади треугольника.

```
Углы равны: 53.13, 36.87, 90.00, периметр равен: 12  
Площадь равна: 210.0, углы равны: 43.60, 46.40, 90.00, периметр равен: 70
```

Рисунок 4 – Результат

5. Выполнение индивидуального задания №1.

В следующих заданиях требуется 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. Реализовать производные классы Complex (комплексное число) и Rational (рациональное число).

```
Ответ: 2 + 8i  
Ответ: 0 + -2i  
Ответ: -14 + 8i  
Ответ: 0.6153846153846154 + -0.07692307692307693i  
Ответ: 33/54  
Ответ: 21/54  
Ответ: 3/54  
Ответ: 27/6
```

Рисунок 5 – Результат

Вывод: мы приобрели навыки по созданию иерархии классов при написании программ с помощью языка программирования Python.

Контрольные вопросы:

1 Что такое наследование как оно реализовано в языке Python?

Наследование в объектно-ориентированных языках программирования позволяет создавать новый класс на основе существующего, наследуя его свойства и методы.

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

Полиморфизм позволяет объектам разных классов использовать одинаковые методы или свойства. В Python полиморфизм реализуется автоматически, благодаря динамической типизации.

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

"Утиная" типизация означает, что важным является не реальное наличие типа объекта, а наличие определенных методов или свойств. Python поддерживает "утиную" типизацию, позволяя вызывать методы, не проверяя явно тип объекта.

4 Каково назначение модуля abc языка программирования Python?

Модуль abc (Abstract Base Classes) предоставляет инструменты для определения абстрактных базовых классов. Абстрактные классы в Python содержат абстрактные методы, которые должны быть реализованы в производных классах.

5 Как сделать некоторый метод класса абстрактным?

Для создания абстрактного метода в Python используется декоратор `@abstractmethod` из модуля abc:

```
from abc import ABC, abstractmethod
```

```
class MyAbstractClass(ABC):  
    @abstractmethod  
    def my_abstract_method(self):  
        pass
```

6 Как сделать некоторое свойство класса абстрактным?

Абстрактные свойства в Python можно определить с использованием модуля abc и декоратора @property.

7 Каково назначение функции isinstance?

Функция isinstance в Python используется для проверки принадлежности объекта к указанному типу. Возвращает True, если объект является экземпляром указанного класса или его подкласса, и False в противном случае.