

РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций
«Наследование и полиморфизм в языке Python»

Отчет по лабораторной работе № 4.3
по дисциплине «Объектно-ориентированное программирование»

Выполнил студент группы ИВТ-б-о-21-1

Шайдеров Дмитрий Викторович.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2023

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

Порядок выполнения работы:

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

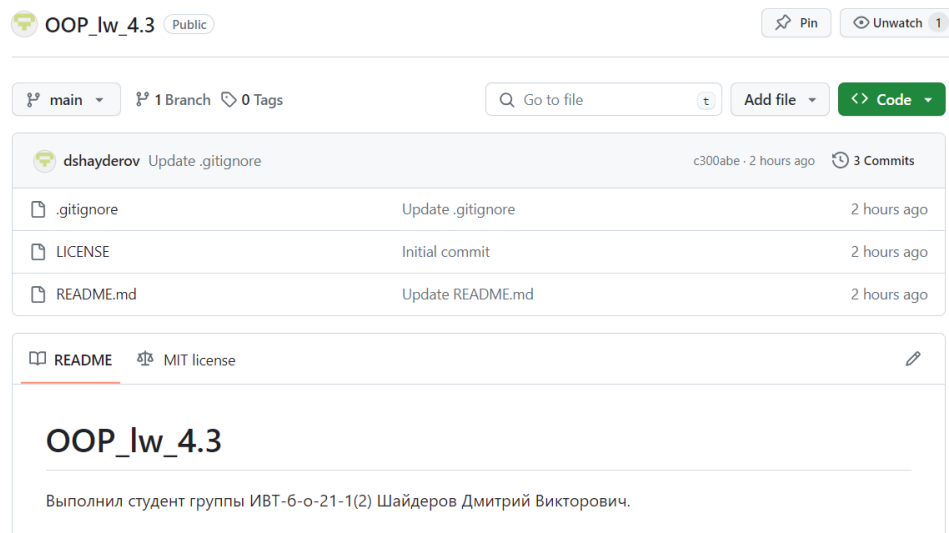


Рисунок 1 - Создание репозитория

2. Выполните клонирование созданного репозитория.

```
C:\Users\Asus\Desktop\Учеба\5 семестр\ООП>git clone https://github.com/dshayderov/OOP_lw_4.3.git
Cloning into 'OOP_lw_4.3'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 11 (delta 2), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (11/11), 4.91 KiB | 4.91 MiB/s, done.
Resolving deltas: 100% (2/2), done.
```

Рисунок 2 - Клонирование репозитория

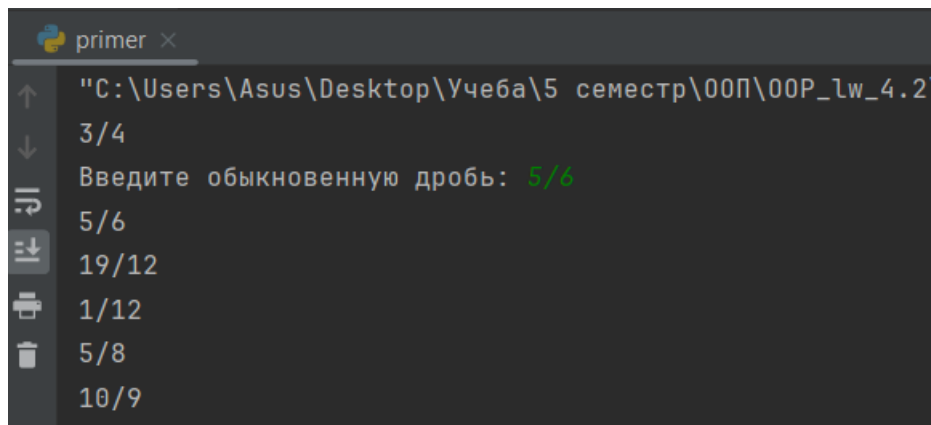
3. Организуйте свой репозиторий в соответствие с моделью ветвления git-flow.

```
C:\Users\Asus\Desktop\Учеба\5 семестр\ООП\ООП_lw_4.3>git checkout -b develop
Switched to a new branch 'develop'

C:\Users\Asus\Desktop\Учеба\5 семестр\ООП\ООП_lw_4.3>
```

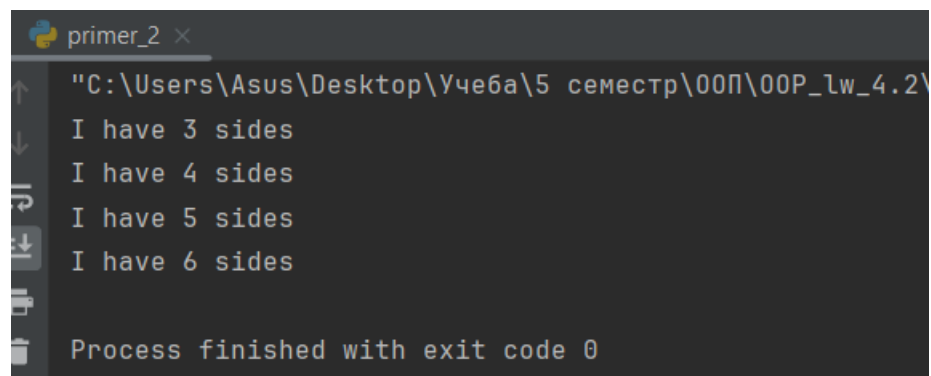
Рисунок 3 - Ветвление по модели git-flow

4. Проработать примеры лабораторной работы.



```
primer x
"C:\Users\Asus\Desktop\Учеба\5 семестр\00П\00Р_lw_4.2\
3/4
Введите обыкновенную дробь: 5/6
5/6
19/12
1/12
5/8
10/9
```

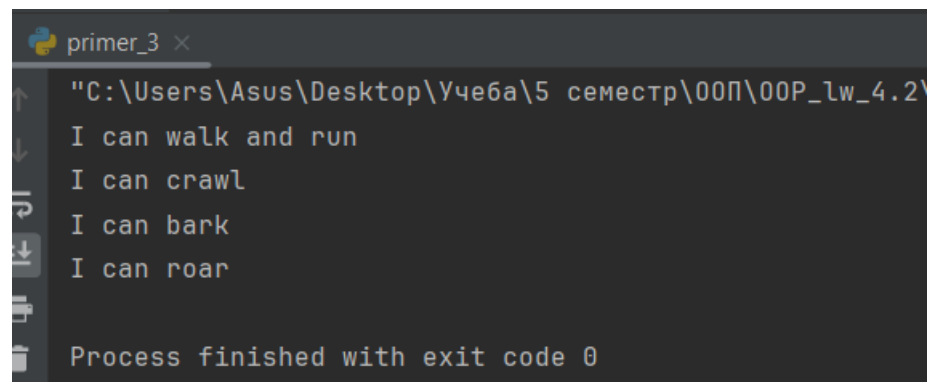
Рисунок 4 - Результат выполнения примера 1



```
primer_2 x
"C:\Users\Asus\Desktop\Учеба\5 семестр\00П\00Р_lw_4.2\
I have 3 sides
I have 4 sides
I have 5 sides
I have 6 sides

Process finished with exit code 0
```

Рисунок 5 - Результат выполнения примера 2



```
primer_3 x
"C:\Users\Asus\Desktop\Учеба\5 семестр\00П\00Р_lw_4.2\
I can walk and run
I can crawl
I can bark
I can roar

Process finished with exit code 0
```

Рисунок 6 - Результат выполнения примера 3

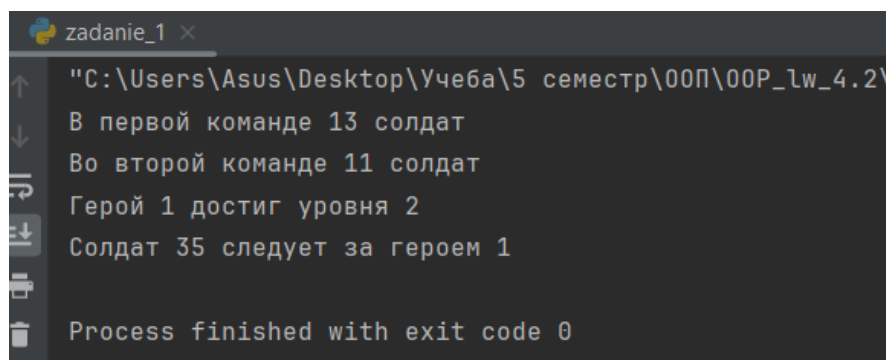
5. Разработайте программу по следующему описанию.

В некой игре-стратегии есть солдаты и герои. У всех есть свойство, содержащее уникальный номер объекта, и свойство, в котором хранится принадлежность команде. У солдат есть метод "иду за героем", который в качестве аргумента принимает объект типа "герой". У героев есть метод увеличения собственного уровня.

В основной ветке программы создается по одному герою для каждой команды. В цикле генерируются объекты-солдаты. Их принадлежность команде определяется случайно. Солдаты разных команд добавляются в разные списки.

Измеряется длина списков солдат противоборствующих команд и выводится на экран. У героя, принадлежащего команде с более длинным списком, увеличивается уровень.

Отправьте одного из солдат первого героя следовать за ним. Выведите на экран идентификационные номера этих двух юнитов.



```
zadanie_1 x
"C:\Users\Asus\Desktop\Учеба\5 семестр\00П\00Р_lw_4.2\
В первой команде 13 солдат
Во второй команде 11 солдат
Герой 1 достиг уровня 2
Солдат 35 следует за героем 1
Process finished with exit code 0
```

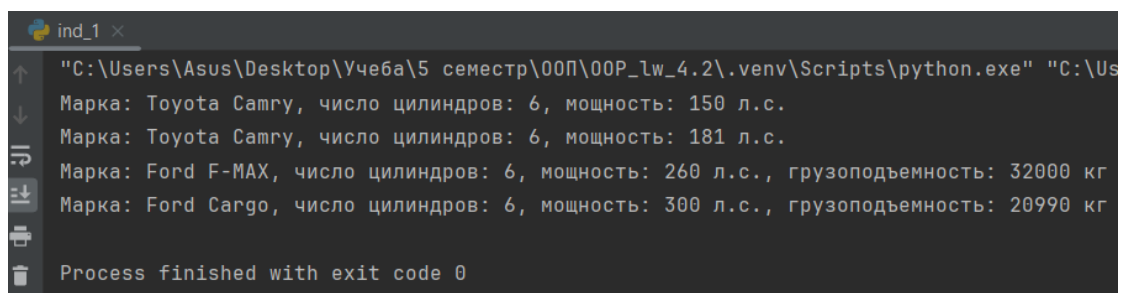
Рисунок 7 - Результат выполнения задания

6. Выполнить индивидуальные задания.

Задание 1. Вариант 21 (1)

Создать базовый класс Car (машина), характеризуемый торговой маркой (строка), числом цилиндров, мощностью. Определить методы переназначения и изменения мощности.

Создать производный класс Lorry (грузовик), характеризуемый также грузоподъемностью кузова. Определить функции переназначения марки и изменения грузоподъемности.

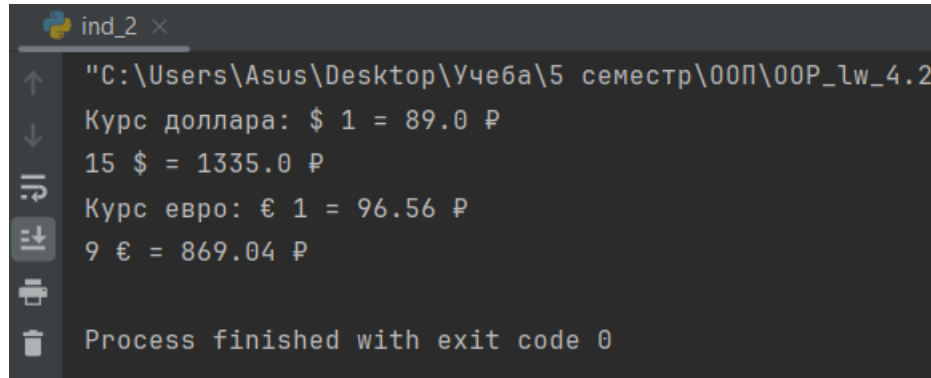


```
ind_1 x
"C:\Users\Asus\Desktop\Учеба\5 семестр\00П\00Р_lw_4.2\.venv\Scripts\python.exe" "C:\Us
Марка: Toyota Camry, число цилиндров: 6, мощность: 150 л.с.
Марка: Toyota Camry, число цилиндров: 6, мощность: 181 л.с.
Марка: Ford F-MAX, число цилиндров: 6, мощность: 260 л.с., грузоподъемность: 32000 кг
Марка: Ford Cargo, число цилиндров: 6, мощность: 300 л.с., грузоподъемность: 20990 кг
Process finished with exit code 0
```

Рисунок 8 - Результат выполнения индивидуального задания 1

Задание 2. Вариант 21 (4)

Создать абстрактный класс Currency (валюта) для работы с денежными суммами. Определить виртуальные функции перевода в рубли и вывода на экран. Реализовать производные классы Dollar (доллар) и Euro (евро) со своими функциями перевода и вывода на экран.



```
ind_2 x
"C:\Users\Asus\Desktop\Учеба\5 семестр\00П\00P_lw_4.2
Курс доллара: $ 1 = 89.0 Р
15 $ = 1335.0 Р
Курс евро: € 1 = 96.56 Р
9 € = 869.04 Р
Process finished with exit code 0
```

Рисунок 9 - Результат выполнения индивидуального задания 2

Контрольные вопросы:

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

Наследование — это когда один класс (подкласс) получает свойства и методы другого класса (суперкласса). Подкласс может наследовать все публичные атрибуты и методы своего суперкласса и добавлять свои собственные. В языке Python наследование реализуется с помощью ключевого слова `class`. Для создания подкласса нужно указать имя суперкласса в скобках после имени подкласса. Подкласс получает все атрибуты и методы суперкласса, их можно использовать напрямую или переопределить.

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

Полиморфизм — это возможность объектов разных классов иметь одно и то же имя метода, но каждый класс может предоставить свою собственную реализацию этого метода. Это позволяет использовать одинаковое имя метода для объектов различных классов, что упрощает программирование и повышает гибкость кода. В языке Python полиморфизм реализуется через наследование и переопределение методов. Если в подклассе метод с тем же именем переопределяется, то при вызове этого метода на объекте подкласса

будет использоваться его реализация, а не реализация суперкласса. Это позволяет использовать одинаковые методы с разным поведением для разных классов.

3. Что такое «утиная» типизация в языке Python?

«Утиная» типизация (англ. duck typing) — это концепция в языке программирования Python, основанная на философии «если она выглядит как утка, плавает как утка и крякает как утка, то это, вероятно, и есть утка». В контексте Python утиная типизация означает, что тип объекта определяется по его возможностям и методам, а не по его явно заданному типу. Иными словами, если объект обладает определенными методами, то мы можем использовать его как экземпляр нужного типа, не задумываясь о его фактическом классе или интерфейсе.

4. Каково назначение модуля abc языка Python?

Модуль abc (аббревиатура от "Abstract Base Classes") является частью стандартной библиотеки языка Python и предоставляет средства для определения абстрактных базовых классов.

5. Как сделать некоторый метод класса абстрактным?

Необходимо декорировать его методы как абстрактные, а реализацию выносить в классы-наследники.

6. Как сделать некоторое свойство класса абстрактным?

Можно потребовать атрибут в конкретных классах, определив их с помощью `@abstractproperty`.

7. Каково назначение функции isinstance?

Функция `isinstance()` проверяет, является ли объект экземпляром указанного класса или его подкласса.

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