

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

Кафедра инфокоммуникаций

«Классы данных в Python»

Отчет по лабораторной работе № 4.6
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 2024

Цель работы: приобретение навыков по работе с классами данных 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.

Owner * Repository name *

dshayderov / OOP_lw_4.6

✓ OOP_lw_4.6 is available.

Great repository names are short and memorable. Need inspiration? How about [supreme-umbrella](#) ?

Description (optional)

☒ Public Anyone on the internet can see this repository. You choose who can commit.

☐ Private You choose who can see and commit to this repository.

Initialize this repository with:

☒ Add a README file This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: Python

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: MIT License

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

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_4.6.git
Cloning into 'OOP_lw_4.6'...
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 соответствии с моделью ветвления git-flow.

```
C:\Users\Asus\Desktop\Учеба\5 семестр\ООП\OOP_lw_4.6>git checkout -b develop
Switched to a new branch 'develop'
```

Рисунок 3 - Ветвление по модели git-flow

4. Проработать пример лабораторной работы

```
C:\Users\Asus\Desktop\Учеба\5 семестр\00П\00Р_lw_4.6\Project>python primer.py
>>> add
Фамилия и инициалы? Сидоров В.Ю.
Должность? Разработчик
Год поступления? 2020
>>> list
```

№	Ф.И.О.	Должность	Год
1	Сидоров В.Ю.	Разработчик	2020

Рисунок 4 - Результат выполнения примера

5. Выполнить индивидуальное задание.

Задание 1.

Выполнить индивидуальное задание лабораторной работы 4.5, используя классы данных, а также загрузку и сохранение данных в формат XML.

Использовать словарь, содержащий следующие ключи: название пункта назначения рейса; номер рейса; тип самолета. Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть размещены в алфавитном порядке по названиям пунктов назначения; вывод на экран пунктов назначения и номеров рейсов, обслуживаемых самолетом, тип которого введен с клавиатуры; если таких рейсов нет, выдать на дисплей соответствующее сообщение.

```
C:\Users\Asus\Desktop\Учеба\5 семестр\00П\00Р_lw_4.6\Project>python ind.py
>>> add
Пункт назначения: Москва
Номер рейса: 123
Тип самолёта: Грузовой
>>> add
Пункт назначения: Новосибирск
Номер рейса: 908
Тип самолёта: Пассажирский
>>> list
```

№	Пункт назначения	Номер рейса	Тип самолета
1	Москва	123	Грузовой
2	Новосибирск	908	Пассажирский

```
>>> select Пассажирский
1: Новосибирск - №908
```

Рисунок 5 - Результат выполнения индивидуального задания

Контрольные вопросы:

1. Как создать класс данных в языке Python?

Класс данных – это класс, который (в основном) хранит данные, хотя на самом деле, нет никаких ограничений. Он разработан при помощи декоратора `@dataclass` следующим образом:

```
from dataclasses import dataclass

@dataclass

class DataClassCard:

    rank: str

    suit: str
```

Декоратор `@dataclass` делает класс – классом данных, прямо над определением класса.

2. Какие методы по умолчанию реализует класс данных?

По умолчанию, классы данных реализует метод `__repr__()`, чтобы предоставить хорошее строковое представление, а также метод `__eq__()`, который в состоянии выполнять базовые сравнения объектов.

3. Как создать неизменяемый класс данных?

Чтобы сделать класс данных неизменяемым, установите `frozen=True` при создании.

```
from dataclasses import dataclass

@dataclass(frozen=True)

class Position:

    name: str

    lon: float = 0

    lat: float = 0
```

Вывод: были приобретены навыки по работе с классами данных при написании программ с помощью языка программирования Python версии 3.x.