РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное

образовательное учреждение высшего

образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

«Работа с исключениями в языке Python»»

Отчет по лабораторной

работе по дисциплине

«Объектно-ориентированное

программирование»

Выполнил студент группы ИВТ-б-о-21-1

Кочкаров Умар Ахматович.

«17» ноября 2023г.

Подпись студента_____

Работа защищена « »_____20__г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2023

Цель работы: приобретение навыков по работе с исключениями при написании программ с помощью языка программирования Python версии 3.x..

Ход работы:

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

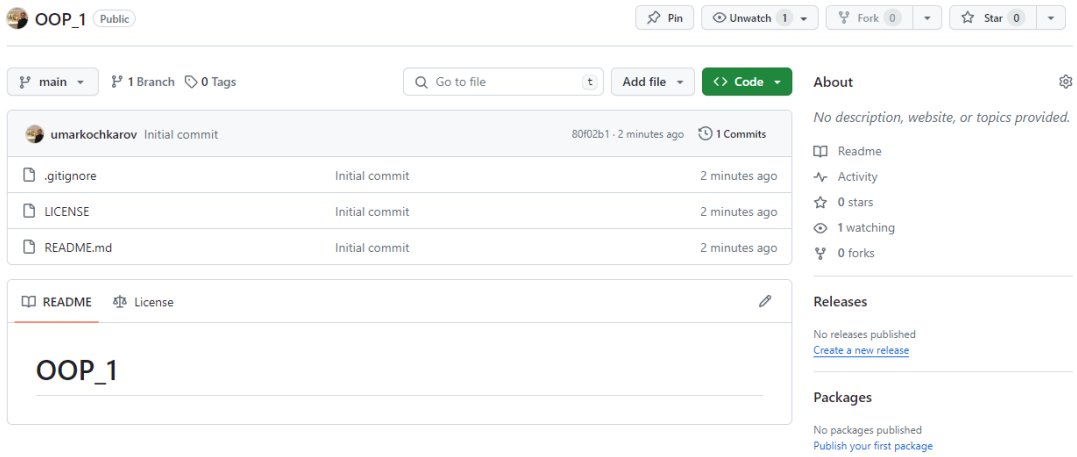

Рисунок 1. Создание репозитория

2. Организовать репозиторий в соответствии с Git-Flow init.

```
erken@LAPTOP-ESTC6OGF MINGW64 ~/Desktop/python/OOP_1 (main)
$ 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:/Users/erken/Desktop/python/OOP_1/.git/hooks]
```

Рисунок 2. Организация в соответствии с Git-Flow init

3. Проработка примеров лабораторной работы:

```
>>> list
+------+----------------------------------+----------------------+----------+
|  №   |              Ф.И.О.              |      Должность       |   Год    |
+------+----------------------------------+----------------------+----------+
|    1 | Kochkarov U.A                    | Director             |     2020 |
|    2 | Shidak A.R.                      | Manager              |     2022 |
+------+----------------------------------+----------------------+----------+
```

Рисунок 3. Пример

4. Индивидуальные задания:

Задание 1. Выполнить индивидуальное задание 1 лабораторной работы 2.19, добавив возможность работы с исключениями и логгирование.

```
C:\Users\erken\AppData\Local\Programs\Python\Python310\python.exe C:/Users/erken/
>>> add
Пункт назначения? Nalchik
Номер самолета? 999
Время отправления? 12:30
>>> add
Пункт назначения? MinVody
Номер самолета? 111
Время отправления? 19:00
>>> list
+------+-------------------------+-----------------+----------------------+
|  №   |     Пункт назначения    | Номер самолета  |  Время отправления   |
+------+-------------------------+-----------------+----------------------+
|    1 | MinVody                 | 111             |                19:00 |
|    2 | Nalchik                 | 999             |                12:30 |
+------+-------------------------+-----------------+----------------------+
>>> select 999
   1: Nalchik
```

Рисунок 4. Индивидуальное задание 1

Задание 2. Изучить возможности модуля logging. Добавить для предыдущего задания вывод в файлы лога даты и времени выполнения пользовательской команды с точностью до миллисекунды.

```
2024-01-29 20:15:48.116 Добавлен поезд №777, пункт назначения: Saint-P, отправляющийся в 01:30
2024-01-29 20:16:19.196 Отображен список поездов.
```

Рисунок 5. Индивидуальное задание 2

Контрольные вопросы:

1. Какие существуют виды ошибок в языке программирования Python?

Синтаксические ошибки, возникающие, если программа написана с нарушением требований Python к синтаксису, и исключения, если в процессе выполнения возникает ошибка.

2. Как осуществляется обработка исключений в языке программирования Python?

Блок кода, в котором возможно появление исключительной ситуации необходимо поместить во внутрь синтаксической конструкции try… except. Если в блоке try возникнет ошибка, программа выполнит блок except.

3. Для чего нужны блоки finnally и else при обработке исключений?

Не зависимо от того, возникнет или нет во время выполнения кода в блоке try исключение, код в блоке finally все равно будет выполнен. Если необходимо выполнить какой-то программный код, в случае если в процессе выполнения блока try не возникло исключений, то можно использовать оператор else.

4. Как осуществляется генерация исключений в языке Python?

Для принудительной генерации исключения используется инструкция raise.

5. Как создаются классы пользовательских исключений в языке Python?

Для реализации собственного типа исключения необходимо создать класс, являющийся наследником от одного из классов исключений.

6. Каково назначение модуля logging?

Для вывода специальных сообщений, не влияющих на функционирование программы, в Python применяется библиотека логов.

Чтобы воспользоваться ею, необходимо выполнить импорт в верхней части файла. С помощью logging на Python можно записывать в лог и исключения.

7. Какие уровни логгирования поддерживаются модулем logging? Приведите примеры, в которых могут быть использованы сообщения с этим уровнем логгирования.

- Debug: самый низкий уровень логирования, предназначенный для отладочных сообщений, для вывода диагностической информации о приложении.

• Info: этот уровень предназначен для вывода данных о фрагментах кода, работающих так, как ожидается.

• Warning: этот уровень логирования предусматривает вывод предупреждений, он применяется для записи сведений о событиях, накоторые программист обычно обращает внимание. Такие события

вполне могут привести к проблемам при работе приложения. Если явно не задать уровень логирования — по умолчанию используется

именно warning.

• Error: этот уровень логирования предусматривает вывод сведений об ошибках — о том, что часть приложения работает не так как

ожидается, о том, что программа не смогла правильно выполниться.

• Critical: этот уровень используется для вывода сведений об очень серьёзных ошибках, наличие которых угрожает нормальному

функционированию всего приложения. Если не исправить такую ошибку — это может привести к тому, что приложение прекратитработу.

Вывод: приобретены навыки по обработке исключений и логгированию при написании программ с использованиемязыка программирования Python версии 3.x.