

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образо-
вательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций
«Работа с исключениями в языке Python»**

**Отчет по лабораторной работе № 4.4
по дисциплине «Объектно-ориентированное программирование»**

Выполнил студент группы ИВТ-б-о-21-1

Богадунов Василий Игоревич.

« » _____ 20__г.

Подпись студента _____

Работа защищена « » _____ 20__г.

Проверил Воронкин Р.А. _____
(подпись)

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

Ход работы:

1. Создание репозитория для выполнения работы.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

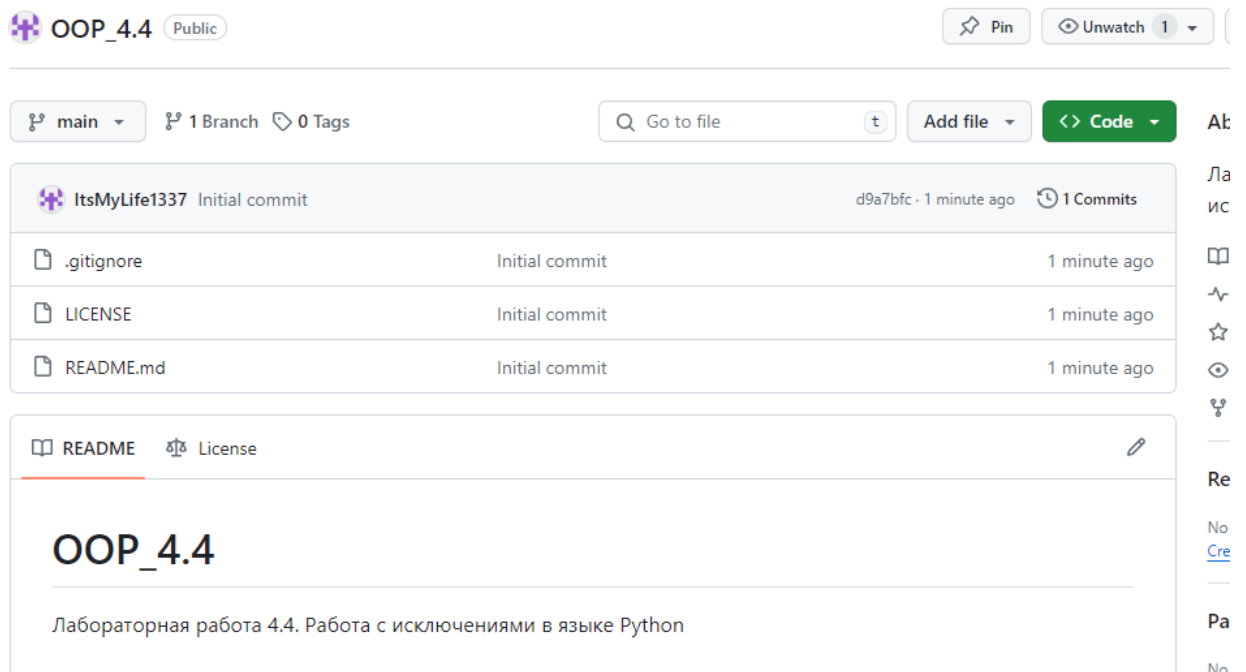
Owner * Repository name *


 ItsMyLife1337 /




Great repository names are short and memorable. Need inspiration? How about [vigilant-palm-tree](#) ?




Description (optional)

Рисунок №1 – Созданный репозиторий



 **OOP_4.4** Public Pin Unwatch 1

 main  1 Branch  0 Tags t Add file <> Code

File	Commit	Time
 .gitignore	Initial commit	1 minute ago
 LICENSE	Initial commit	1 minute ago
 README.md	Initial commit	1 minute ago

README License

OOP_4.4

Лабораторная работа 4.4. Работа с исключениями в языке Python

Рисунок №2 – Созданный репозиторий

```

C:\Users\Admin>cd /d c:\users\admin\desktop\git

c:\Users\Admin\Desktop\git>git clone https://github.com/ItsMyLife1337/OOP_4.4.git
Cloning into 'OOP_4.4'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.

c:\Users\Admin\Desktop\git>cd /d c:\users\admin\desktop\git\OOP4.4
Системе не удается найти указанный путь.

c:\Users\Admin\Desktop\git>cd /d c:\users\admin\desktop\git\OOP_4.4

c:\Users\Admin\Desktop\git\OOP_4.4>

```

Рисунок 3 – Клонирование репозитория

```

c:\Users\Admin\Desktop\git\OOP_4.4>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/Admin/Desktop/git/OOP_4.4/.git/hooks]

c:\Users\Admin\Desktop\git\OOP_4.4>_

```

Рисунок 4 – Организация репозитория в соответствии с моделью ветвления git flow

2. Решите следующую задачу: напишите программу, которая запрашивает ввод двух значений. Если хотя бы одно из них не является числом, то должна выполняться конкатенация, т. е. соединение, строк. В остальных случаях введенные числа суммируются.

```

PS C:\Users\Admin\Desktop\git\OOP_4.4\Tasks> & C:/Users/Admin/AppData/
/python3.11.exe c:/Users/Admin/Desktop/git/OOP_4.4/Tasks/Task1.py
1
2
Сумма чисел: 12.0
PS C:\Users\Admin\Desktop\git\OOP_4.4\Tasks> & C:/Users/Admin/AppData/
/python3.11.exe c:/Users/Admin/Desktop/git/OOP_4.4/Tasks/Task1.py
1a
2
Конкатенация 1a2
PS C:\Users\Admin\Desktop\git\OOP_4.4\Tasks> 

```

Рисунок 5 – Результат выполнения программы

3. Решите следующую задачу: напишите программу, которая будет генерировать матрицу из случайных целых чисел. Пользователь может указать число строк и столбцов, а также диапазон целых чисел. Произведите обработку ошибок ввода пользователя

```

/python3.11.exe c:/Users/Admin/Desktop/git/OOP_4.4/Tasks/Task2.py
2
2ф
Ошибка ввода! Вводите только целые числа!
PS C:\Users\Admin\Desktop\git\OOP_4.4\Tasks> & C:/Users/Admin/AppData/Local/M
/python3.11.exe c:/Users/Admin/Desktop/git/OOP_4.4/Tasks/Task2.py
Введите число строк: 2
1
9
[[9, 7, 9], [6, 6, 1]]
PS C:\Users\Admin\Desktop\git\OOP_4.4\Tasks> & C:/Users/Admin/AppData/Local/M
/python3.11.exe c:/Users/Admin/Desktop/git/OOP_4.4/Tasks/Task2.py

```

Рисунок 6 – Результат работы программы

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

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

```

c:\Users\Admin\Desktop\git\OOP_4.7>cd /d c:\users\admin\desktop\git\OOP_4.4
c:\Users\Admin\Desktop\git\OOP_4.4>cd /d c:\users\admin\desktop\git\OOP_4.4\Tasks
c:\Users\Admin\Desktop\git\OOP_4.4\Tasks>python individual1.py add test.json --name="Arsen QQ" --group="20" --grade="5 4 4 5 4"
c:\Users\Admin\Desktop\git\OOP_4.4\Tasks>python individual1.py display "C:\Users\Admin\test.json"
+-----+-----+-----+-----+
| № | Ф.И.О. | Группа | Успеваемость |
+-----+-----+-----+-----+
| 1 | Arsen QQ | 20 | 5 4 4 5 4 |
+-----+-----+-----+-----+
c:\Users\Admin\Desktop\git\OOP_4.4\Tasks>_

```

Рисунок 7 – Результат выполнения

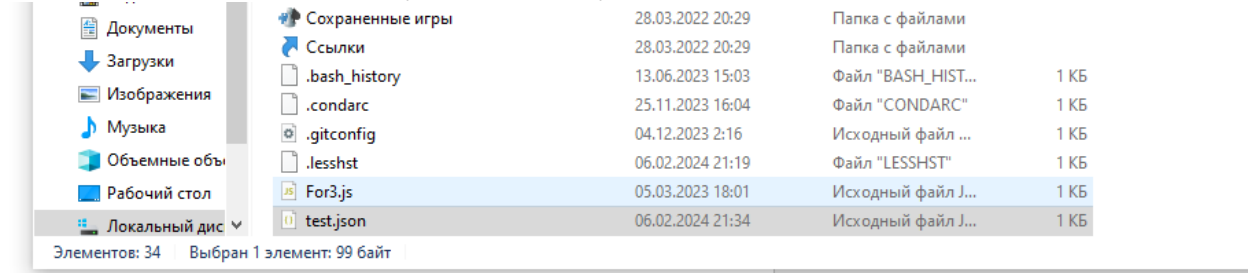


Рисунок 8 – Полученный файл

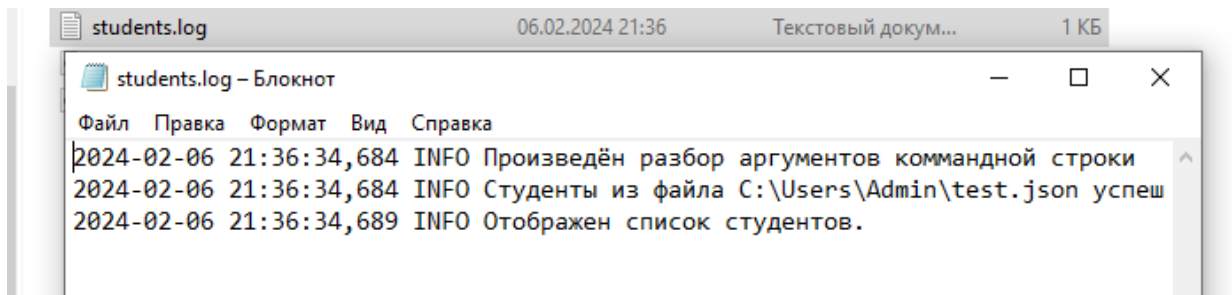


Рисунок 9 – Logging

Вывод: в результате выполнения лабораторной работы были приобретены навыки по работе с исключениями при написании программ с помощью языка программирования Python версии 3.x.

Ответы на 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 `finally` 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 логгирования, 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 логгирования предусматривает вывод предупреждений, он применяется для записи сведений о событиях, на которые программист обычно обращает внимание. Такие события вполне могут привести к проблемам при работе приложения. Если явно не задать уровень логгирования — по умолчанию используется именно warning.

- **Error**: этот уровень логгирования 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 прекратит работу.