

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное  
образовательное учреждение высшего образования**

**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**Отчет по лабораторной работе №4.4**

**Работа с исключениями в языке Python**

**по дисциплине «Технологии программирования и алгоритмизации»**

Выполнил студент группы ИВТ-б-о-20-1

Хашиев Х.М. « » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 20\_\_ г.

Проверил Воронкин Р.А. \_\_\_\_\_

(подпись)

Ставрополь 2022

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

### Ход работы: Примеры

<https://github.com/Mirror-Shard/L4.4>

1. Изучил теоретический материал и приступил к выполнению примеров:

```
(tools) C:\Users\1\Desktop\Алгоритмизация\Lab12\L2.17\examples>python example_1.py add workers.json --name="anton" --post="president" --year=2002
```

Рисунок 1 – Добавление работника в новый файл

```
(tools) C:\Users\1\Desktop\Алгоритмизация\Lab12\L2.17\examples>python example_1.py display workers.json
```

№	Ф.И.О.	Должность	Год
1	anton	president	2002

Рисунок 2 – Вывод информации о работниках из файла

### Задание 1

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.

1. Написал программу, ввёл сначала два числа, затем символ и число:

```
Первое значение: 5
Второе значение: 6
Результат: 11
Первое значение: j
Второе значение: 3
Результат: j3
```

Рисунок 2 – Работа программы

## Задание 2

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.

2. Ввёл количество столбцов, строк и диапазон чисел. Программа успешно сгенерировала матрицу:

```
Введите количество столбцов: 4
Введите количество строк: 3
Введите два числа – от и до: 1 10
[5, 6, 8, 2]
[9, 7, 9, 1]
[4, 6, 4, 8]
```

Рисунок 3 – Генерация матрицы

## Индивидуальное задание 1

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

1. Добавил обработку ошибки при загрузке не существующего файла:

```
(base) C:\Users\1\Desktop\ООП\Лабораторная 4.4\Lab_4.4\individual>python ind_1.py add data.json --name="anton" --group=1
9 --average_estimation=4.8
WARNING:root:Файл не найден, создается новый
```

Рисунок 4 – Обработка ошибки

## Индивидуальное задание 2

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

2. Перевёл логгирование в файл logs.log, добавил время:

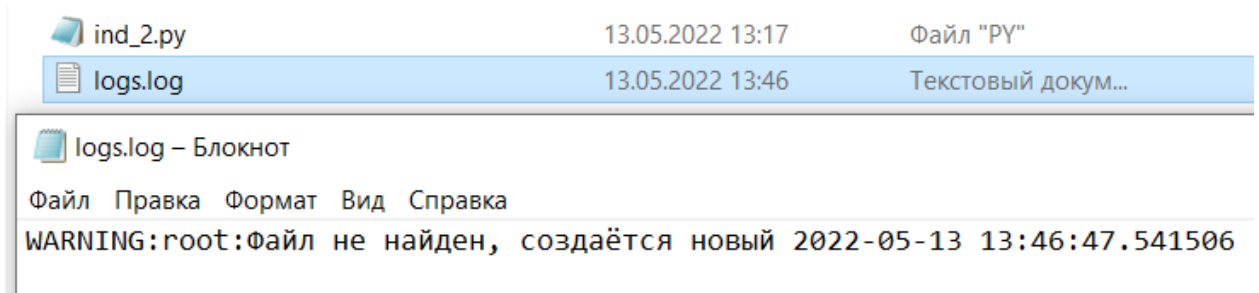


Рисунок 5 – Логгирование с датой

### Контрольные вопросы:

1. Какие существуют виды ошибок в языке программирования Python?

- SystemExit;
- KeyboardInterrupt;
- GeneratorExit;
- Exception;
- StopIteration;
- StopAsyncIteration;
- ArithmeticError;
- FloatingPointError;
- OverflowError;
- ZeroDivisionError;
- AssertionError;
- AttributeError;
- BufferError;
- EOFError;
- ImportError;
- ModuleNotFoundError;
- LookupError;
- IndexError;
- KeyError;
- MemoryError;
- NameError;

- UnboundLocalError;
- OSError;
- BlockingIOError;
- ChildProcessError;
- ConnectionError;
- BrokenPipeError;
- ConnectionAbortedError;
- ConnectionRefusedError;
- ConnectionResetError;
- FileExistsError;
- FileNotFoundError;
- InterruptedError;
- IsADirectoryError;
- NotADirectoryError;
- PermissionError;
- ProcessLookupError;
- TimeoutError;
- ReferenceError;
- RuntimeError;
- NotImplementedError;
- RecursionError;
- SyntaxError;
- IndentationError;
- TabError;
- SystemError;
- TypeError;
- ValueError;
- UnicodeError;
- UnicodeDecodeError;
- UnicodeEncodeError;

- UnicodeTranslateError;
- Warning;
- DeprecationWarning;
- PendingDeprecationWarning;
- RuntimeWarning;
- SyntaxWarning;
- UserWarning;
- FutureWarning;
- ImportWarning;
- UnicodeWarning;
- BytesWarning;
- ResourceWarning.

2. Как осуществляется обработка исключений в языке программирования Python?

Обработка исключений нужна для того, чтобы приложение не завершалось аварийно каждый раз, когда возникает исключение. Для этого блок кода, в котором возможно появление исключительной ситуации необходимо поместить во внутрь синтаксической конструкции `try... 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 пользовательский 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 журналирования.

DEBUG:root:Debug message!

INFO:root:Info message!

WARNING:root:Warning message!

ERROR:root>Error message!

CRITICAL:root:Critical message!

**Вывод:** в ходе работы приобрёл навыки по работе с исключениями с помощью языка программирования Python версии 3.