

**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 № 4.4**  
**по дисциплине «Объектно-ориентированное программирование»**

Выполнил студент группы ИВТ-б-о-21-1

Стригалов Дмитрий.

«    » \_\_\_\_\_ 20\_\_ г.

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.

**Порядок выполнения работы:**

### **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.

```
E:\Python\4.4\.venv\Scripts\python.exe E:\Python\4.4\zadanie1.py
Первое значение: 1
Второе значение: a
1a
```

Рисунок 1 - Результат выполнения задания 1

### **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.

```
Строки: 3
Столбцы: 3
Начало диапазона целых чисел: 4
Конец диапазона целых чисел: 7

6 4 6
4 7 7
6 5 5
```

Рисунок 2- Результат выполнения задания 2

### **Индивидуальное задание.**

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

```
INFO:root:Добавлен магазин: H&M,и Кроссовки, С ценой 1000 рублей.
INFO:root:Отображен список магазинов.
```

**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.

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