

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
Работа с исключениями в языке Python

по дисциплине «Объектно-ориентированное программирование»

Выполнил студент группы ИВТ-б-о-20-1

Колбасов В.С. « » _____ 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

1. Изучив методические указания, приступил к разбору примера.

```
C:\Users\vaaaa\anaconda3\envs\1\python.exe "C:/Users/vaaaa/Desktop/ООП/4/Пример 1.py"
>>> list
+-----+-----+-----+-----+
| No |          Ф.И.О.          |      Должность      |   Год   |
+-----+-----+-----+-----+
>>> add
Фамилия и инициалы? Колбасов В.С.
Должность? студент
Год поступления? 2019
>>> list
+-----+-----+-----+-----+
| No |          Ф.И.О.          |      Должность      |   Год   |
+-----+-----+-----+-----+
|  1 | Колбасов В.С.          | студент             |  2019   |
+-----+-----+-----+-----+
>>>
```

Рисунок 4.1 – Проверка правильности работы кода 2.

Затем начал выполнять задания для моего варианта.

```

class Summ():
    def __init__(self, inp1, inp2):
        self.a = inp1
        self.b = inp2

    def summ(self):
        print("Сумма чисел: ", int(self.a) + int(self.b))

    def cont(self):
        print("Результат конкатенации: ", self.a + self.b)

def main():
    try:
        inp1 = input("Введите первое число: ")
        inp2 = input("Введите второе число: ")
        summ = Summ(inp1, inp2)
        summ.summ()
    except ValueError as v:
        summ = Summ(inp1, inp2)
        summ.cont()

if __name__ == "__main__":
    main()

```

Рисунок 4.2 – Код первого индивидуального задания

```

C:\Users\vaaaa\anaconda3\envs\1\python.exe C:/Users/vaaaa/Desktop/00П/4/3дн1.py
Введите первое число: 4
Введите второе число: 5
Сумма чисел: 9

```

Рисунок 4.3 – Проверка кода первого задания

```

class Matrix:
    def __init__(self, inp1, inp2, inp3, inp4):
        self.strok = int(inp1)
        self.stolb = int(inp2)
        self.min = int(inp3)
        self.max = int(inp4)

    def random_init(self):
        pprint([[random.randrange(self.min, self.max) for y in range(self.stolb)] for x in range(self.strok)])

def main():
    try:
        inp1 = input("Введите количество строк: ")
        inp2 = input("Введите количество столбцов: ")
        inp3 = input("Введите минимальную границу диапазона чисел: ")
        inp4 = input("Введите максимальную границу диапазона чисел: ")
        mat = Matrix(inp1, inp2, inp3, inp4)
        mat.random_init()
    except ValueError as v:
        print("Ошибка при вводе значения!")

```

Рисунок 4.4 – Код второго задания

```

C:\Users\vaaaa\anaconda3\envs\1\python.exe C:/Users/vaaaa/Desktop/00П/4/3дн2.py
Введите количество строк: 4
Введите количество столбцов: 5
Введите минимальную границу диапазона чисел: 2
Введите максимальную границу диапазона чисел: 30
[[19, 20, 20, 22, 11],
 [16, 24, 26, 7, 27],
 [15, 21, 14, 26, 21],
 [28, 22, 17, 15, 14]]

```

Рисунок 4.5 – Проверка кода второго задания

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 выполнения лабораторной работы были приобретены простейшие навыки по работе с исключениями в языке программирования Python.