МИНИCTEPCTBO 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 |
| Павленко М.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.

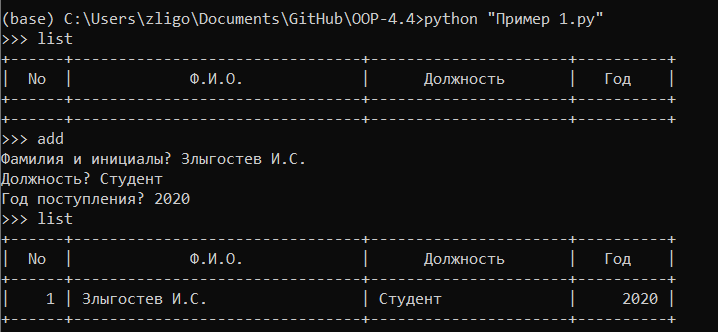


Рисунок 4.1 – Проверка правильности работы кода

1. Затем начал выполнять задания для моего варианта.

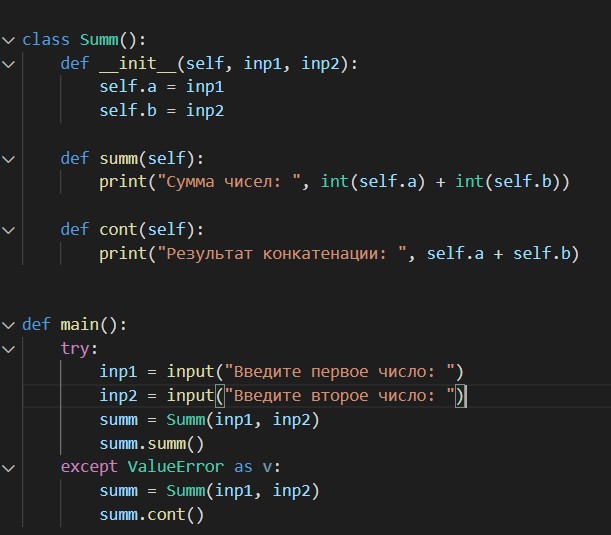


Рисунок 4.2 – Код первого индивидуального задания

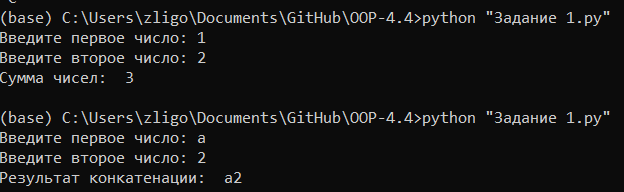


Рисунок 4.3 – Проверка кода первого задания

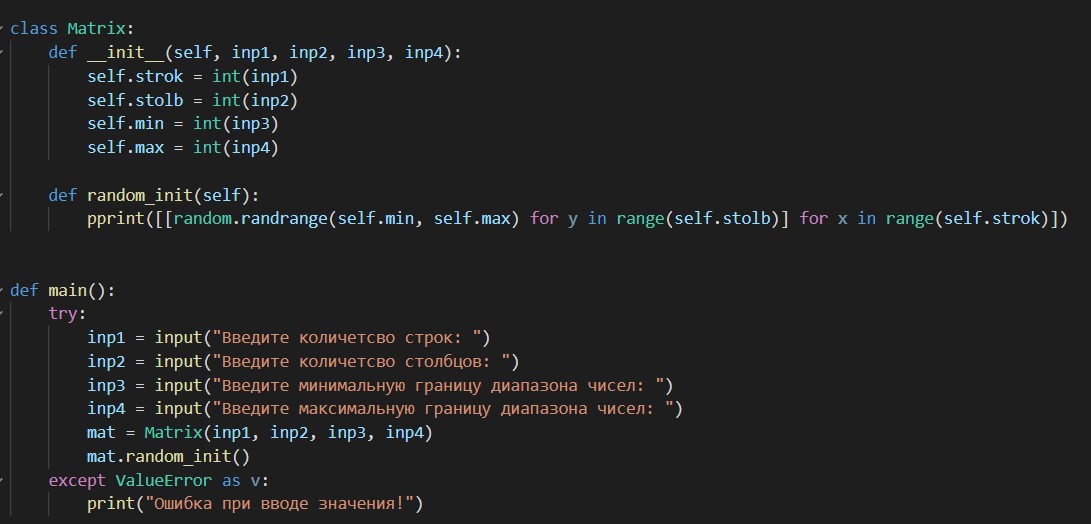


Рисунок 4.4 – Код второго задания

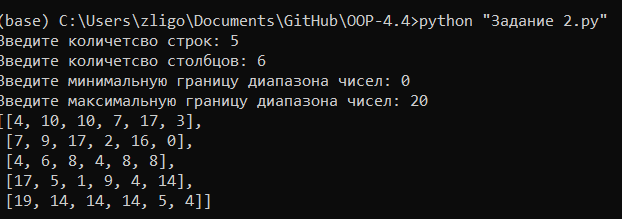


Рисунок 4.5 – Проверка кода второго задания Контрольные вопросы

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.

1. 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.

1. 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.

1. Как осуществляется генерация исключений в языке Python?

Для принудительной генерации исключения используется инструкция

raise.

1. Как создаются классы пользовательский исключений в языке

Python?

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

1. 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.

1. 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.