

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

Отчет по лабораторной работе № 4.4
Работа с исключениями в языке Python

Выполнил студент группы ИВТ-б-о-20-1

Ищенко М.А.

Работа защищена « » _____ 20__ г.

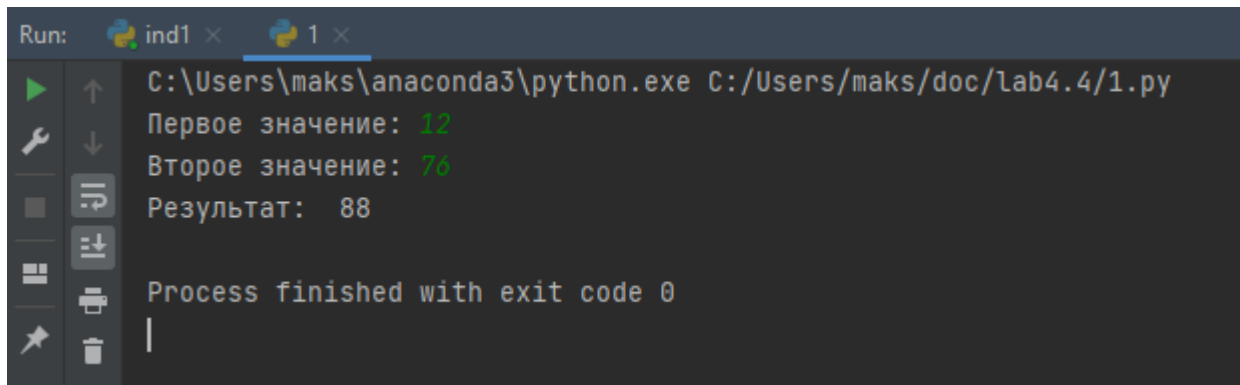
Проверил(а) _____

Ставрополь 2022

Цель работы: приобретение навыков по созданию иерархии классов при написании программ с помощью языка программирования Python версии 3.x.

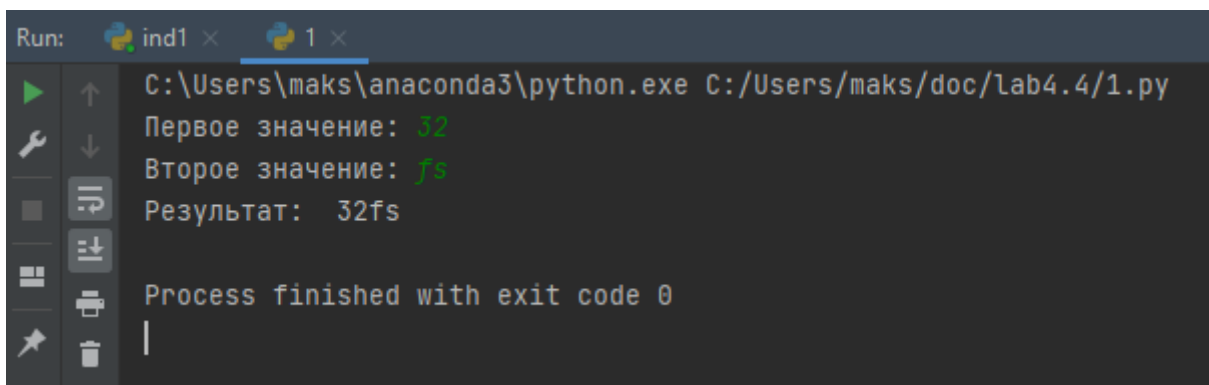
Создан общедоступный репозиторий на GitHub. Дополнен файл .gitignore необходимыми правилами для работы с IDE PyCharm.

Выполнено общее задание 1, рис. 1-2



```
Run: ind1 x 1 x
C:\Users\maks\anaconda3\python.exe C:/Users/maks/doc/lab4.4/1.py
Первое значение: 12
Второе значение: 76
Результат: 88
Process finished with exit code 0
```

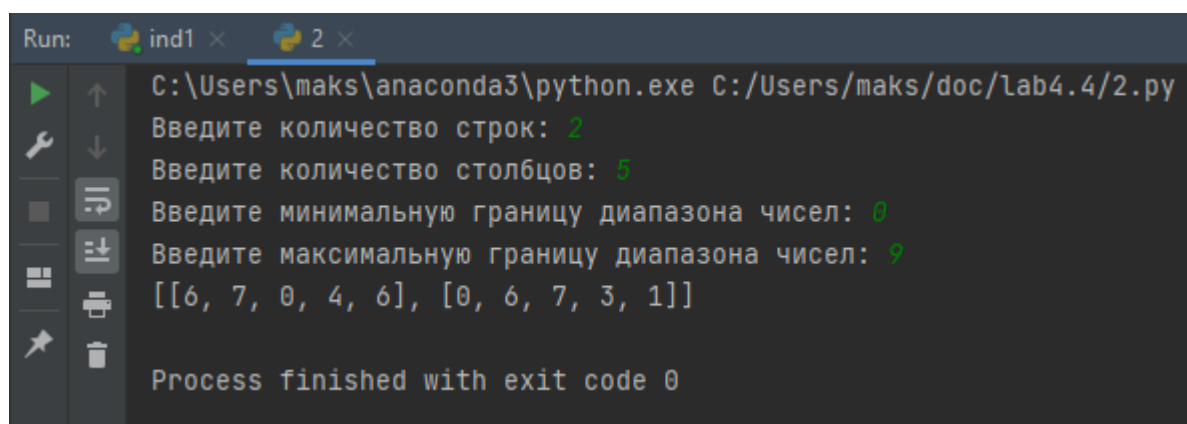
Рисунок 1



```
Run: ind1 x 1 x
C:\Users\maks\anaconda3\python.exe C:/Users/maks/doc/lab4.4/1.py
Первое значение: 32
Второе значение: fs
Результат: 32fs
Process finished with exit code 0
```

Рисунок 2

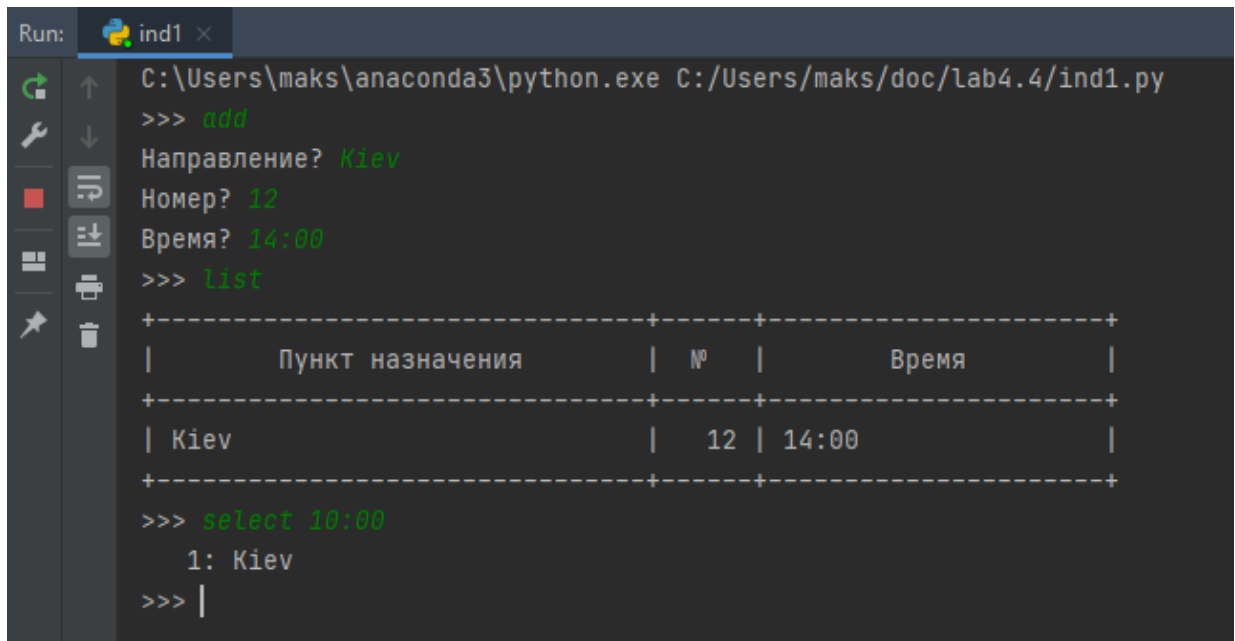
Выполнено общее задание 2, рис. 3



```
Run: ind1 x 2 x
C:\Users\maks\anaconda3\python.exe C:/Users/maks/doc/lab4.4/2.py
Введите количество строк: 2
Введите количество столбцов: 5
Введите минимальную границу диапазона чисел: 0
Введите максимальную границу диапазона чисел: 9
[[6, 7, 0, 4, 6], [0, 6, 7, 3, 1]]
Process finished with exit code 0
```

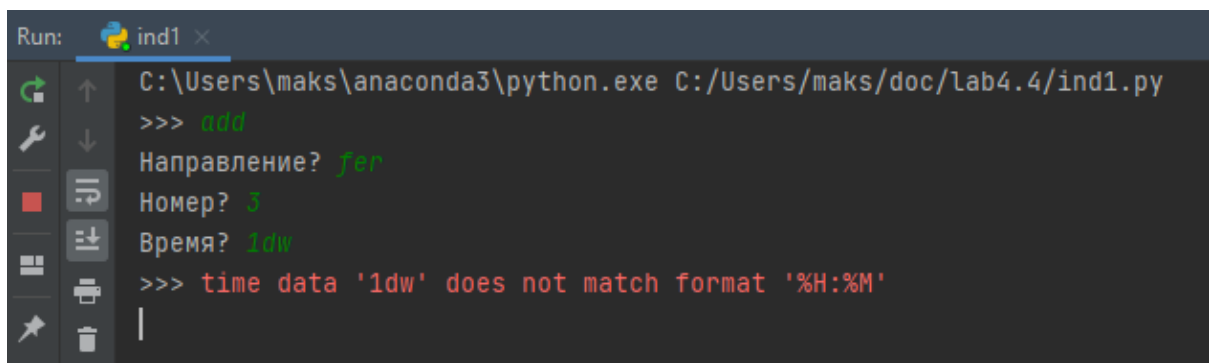
Рисунок 3

Выполнено первое индивидуальное задание варианта 8, рис. 4-6



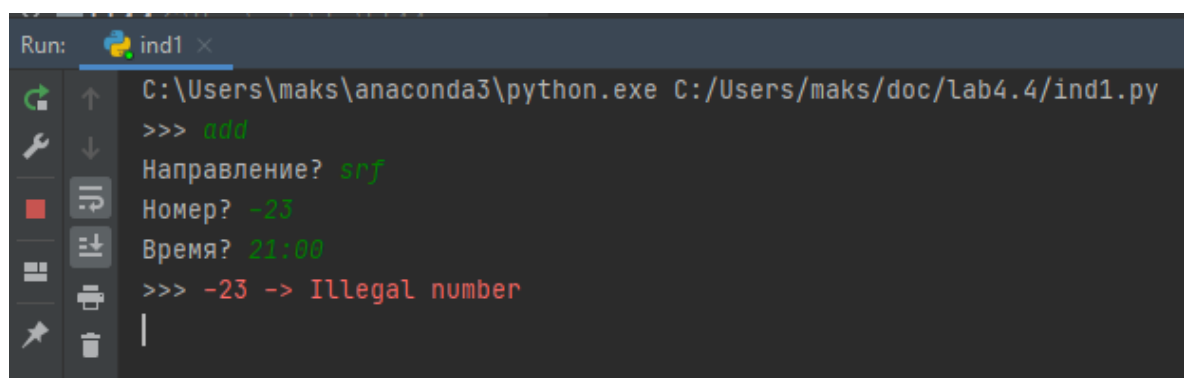
```
Run: ind1 x
C:\Users\maks\anaconda3\python.exe C:/Users/maks/doc/lab4.4/ind1.py
>>> add
Направление? Kiev
Номер? 12
Время? 14:00
>>> list
+-----+-----+-----+
| Пункт назначения | № |      Время      |
+-----+-----+-----+
| Kiev             | 12 | 14:00           |
+-----+-----+-----+
>>> select 10:00
1: Kiev
>>> |
```

Рисунок 4



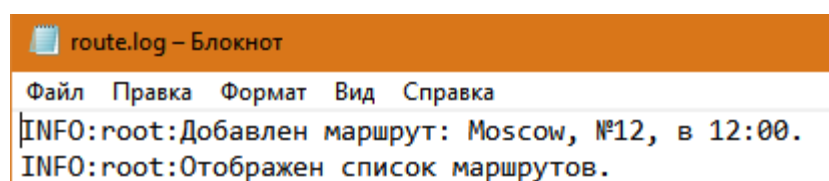
```
Run: ind1 x
C:\Users\maks\anaconda3\python.exe C:/Users/maks/doc/lab4.4/ind1.py
>>> add
Направление? fer
Номер? 3
Время? 1dw
>>> time data '1dw' does not match format '%H:%M'
|
```

Рисунок 5



```
Run: ind1 x
C:\Users\maks\anaconda3\python.exe C:/Users/maks/doc/lab4.4/ind1.py
>>> add
Направление? srf
Номер? -23
Время? 21:00
>>> -23 -> Illegal number
|
```

Рисунок 6



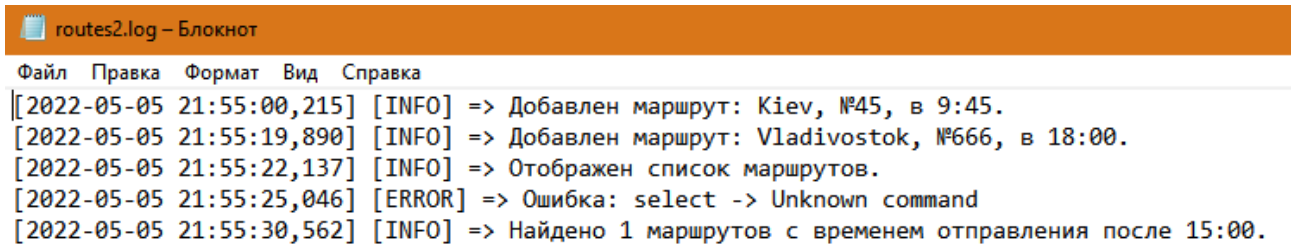
route.log – Блокнот

Файл Правка Формат Вид Справка

INFO:root:Добавлен маршрут: Moscow, №12, в 12:00.
INFO:root:Отображен список маршрутов.

Рисунок 7

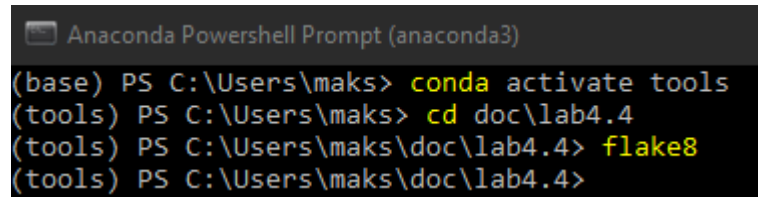
Выполнено второе индивидуальное задание варианта 8, рис. 8



```
routes2.log – Блокнот
Файл  Правка  Формат  Вид  Справка
[2022-05-05 21:55:00,215] [INFO] => Добавлен маршрут: Kiev, №45, в 9:45.
[2022-05-05 21:55:19,890] [INFO] => Добавлен маршрут: Vladivostok, №666, в 18:00.
[2022-05-05 21:55:22,137] [INFO] => Отображен список маршрутов.
[2022-05-05 21:55:25,046] [ERROR] => Ошибка: select -> Unknown command
[2022-05-05 21:55:30,562] [INFO] => Найдено 1 маршрутов с временем отправления после 15:00.
```

Рисунок 8

Программы проверены на flake8, рис. 9



```
Anaconda Powershell Prompt (anaconda3)
(base) PS C:\Users\maks> conda activate tools
(tools) PS C:\Users\maks> cd doc\lab4.4
(tools) PS C:\Users\maks\doc\lab4.4> flake8
(tools) PS C:\Users\maks\doc\lab4.4>
```

Рисунок 9

Контрольные вопросы:

1. Какие существуют виды ошибок в языке программирования Python?

- SystemExit;
- KeyboardInterrupt;
- GeneratorExit;
- Exception;
 - StopIteration;
 - StopAsyncIteration;
 - ArithmeticError;
 - FloatingPointError;
 - и т.д.

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 занятия были приобретены навыки по перегрузке операторов при написании программ с помощью языка программирования Python версии 3.x.