Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт перспективной инженерии
Департамент цифровых, робототехнических систем и электроники

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №4
дисциплины «Объектно-ориентированное программирование»
Вариант 3

Выполнил:
Говоров Егор Юрьевич
3 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем»,
очная форма обучения

_____
(подпись)

Проверил:
Воронкин Роман Александрович,
доцент департамента цифровых,
робототехнических систем и
электроники

_____
(подпись)

Отчет защищен с оценкой_____Дата защиты_____

Ставрополь, 2024 г.

Тема: Работа с исключениями в языке Python

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

Ссылка на репозиторий: https://github.com/Artorias1469/object-oriented-programming_4.git

Ход работы:

1. Проработал пример:

```
D:\Users\Admin\Рабочий стол\ООП\object-oriented-programming_4\Prog>python Prim_1.py
>>> add
Фамилия и инициалы? Говоров Егор Юрьевич
Должность? Инженер
Год поступления? 2015
>>> list
+------+--------------------------------+----------------------+----------+
|  №   |             Ф.И.О.             |      Должность       |   Год    |
+------+--------------------------------+----------------------+----------+
|    1 | Говоров Егор Юрьевич           | Инженер              |     2015 |
+------+--------------------------------+----------------------+----------+
>>> save workers.xml
>>> load workers.xml
>>> select 5
'>=' not supported between instances of 'int' and 'str'
>>> exit

D:\Users\Admin\Рабочий стол\ООП\object-oriented-programming_4\Prog>
```

Рис 1. Вывод выполнения кода из примера 1

2. Решение задач в ходе выполнения:

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.

```
D:\Users\Admin\Рабочий стол\ООП\object-oriented-programming_4\Prog>python IDZ.py
Введите значение 4
Введите значение 5
9.0

D:\Users\Admin\Рабочий стол\ООП\object-oriented-programming_4\Prog>python IDZ.py
Введите значение a
Введите значение 9
a9
```

Рис 2. Выполнение программы Task_one.py

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.

```
D:\Users\Admin\Рабочий стол\ООП\object-oriented-programming_4\Prog>python Task_two.py
[5, 1, 3, 3, 4]
[3, 2, 5, 2, 2]
[2, 5, 4, 1, 1]
[1, 4, 5, 2, 2]
```

Рис 3. Выполнение программы Task_two.py

3. Выполнение индивидуального задания:

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

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

```
(base) D:\Users\Admin\Рабочий стол\ООП\object-oriented-programming_4\Prog>python Ind.py -a
Введите название пункта назначения: Москва-Ставрополь
Введите номер рейса: 244
Введите тип самолета: Грузовой

(base) D:\Users\Admin\Рабочий стол\ООП\object-oriented-programming_4\Prog>python Ind.py -p
+-------------------------------+----------------------+-----------------+
|   Название пункта назначения  |     Номер рейса      |  Тип самолета   |
+-------------------------------+----------------------+-----------------+
| Москва-Сочи                   | 12                   | Грузовой        |
| Москва-Ставрополь             | 244                  | Грузовой        |
| Москва-Ставрополь             | 244                  | Грузовой        |
+-------------------------------+----------------------+-----------------+
```

Рис 4. Результат выполнения Ind.py

```
2024-10-29 23:19:56.586 - INFO - Загружены данные из файла C:\Users\Admin\flights.json.
2024-10-29 23:19:56.587 - INFO - Сохранены данные о рейсах в файл C:\Users\Admin\flights.json.
2024-10-29 23:19:56.587 - INFO - Добавлен рейс 244 в пункт Москва-Ставрополь с типом самолета Грузовой.
2024-10-29 23:20:05.356 - INFO - Загружены данные из файла C:\Users\Admin\flights.json.
2024-10-29 23:20:05.357 - INFO - Вывод списка всех рейсов.
```

Рис 5. Файл лог с датой и временем с точностью до миллисекунды

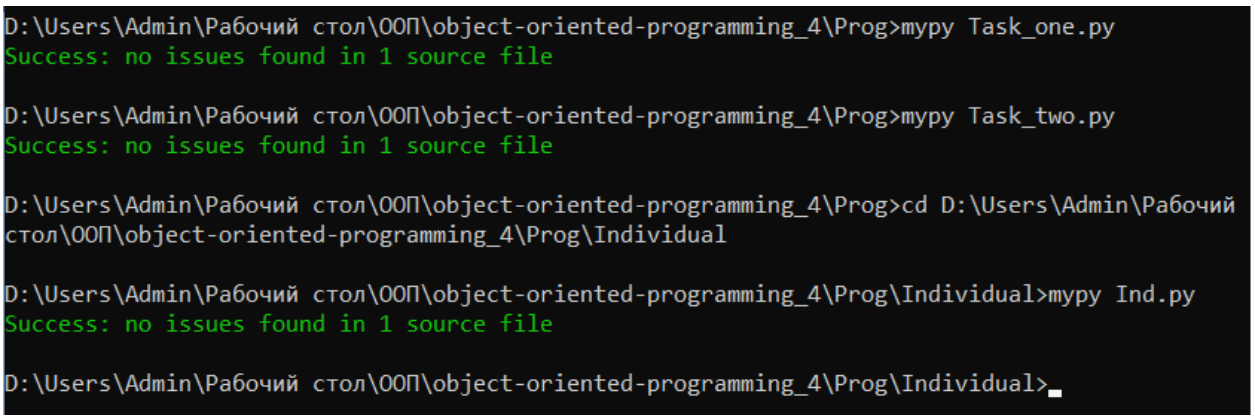

Рис 6. Иллюстрация работы утилиты MyPy

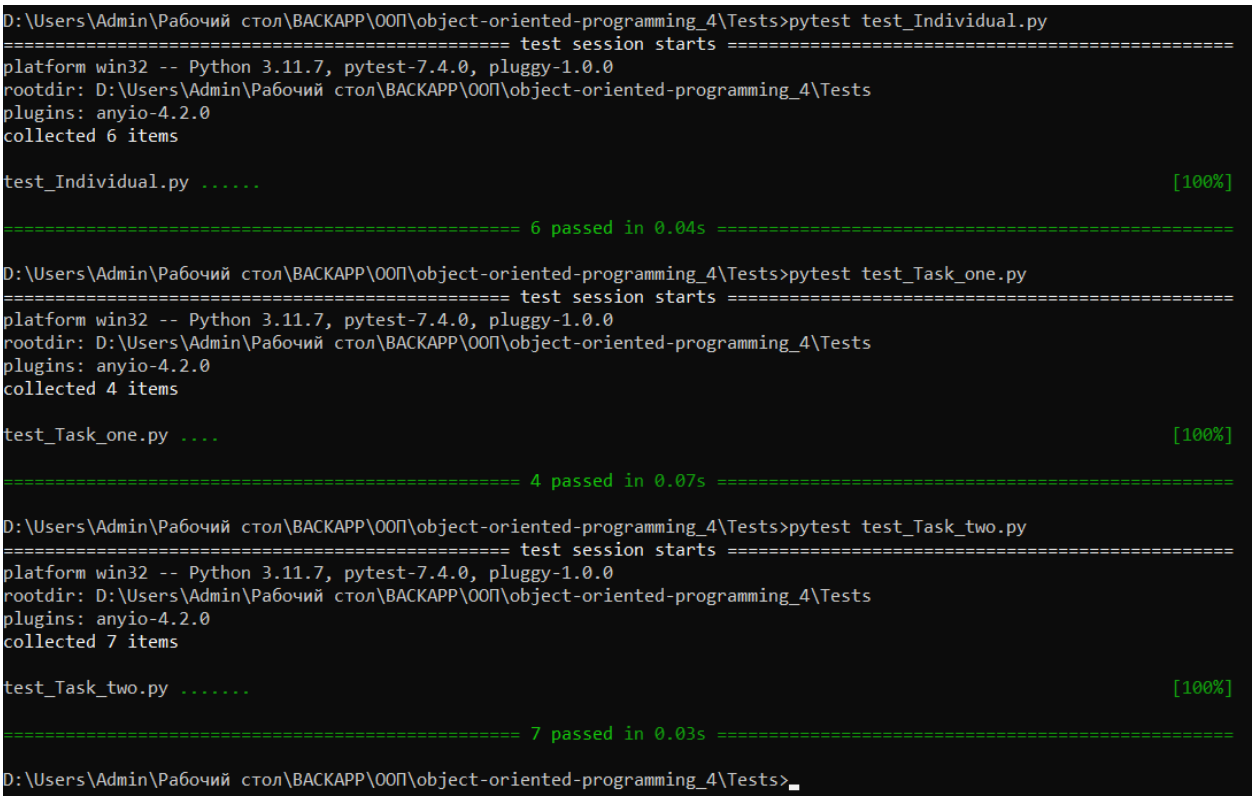

Рис 7. Добавил фреймворк pytest для проверки кода на корректность путем получения ожидаемого результата

Ответы на контрольные вопросы:

1. Какие существуют виды ошибок в языке программирования Python?

В Python существуют два основных вида ошибок:

Синтаксические ошибки: возникающие при нарушении синтаксиса языка и определяющиеся на этапе парсинга программы.

Исключения: ошибки, возникающие во время выполнения программы при корректном синтаксисе кода, такие как деление на ноль, отсутствие файла и т.п.

2. Как осуществляется обработка исключений в языке программирования Python?

Обработка исключений в Python выполняется с помощью блока try…except. Код, который может вызвать исключение, помещается в блок try, а возможные исключения обрабатываются в блоке except. Это позволяет программе продолжать выполнение после обработки ошибки.

3. Для чего нужны блоки finally и else при обработке исключений?

finally: выполняется в любом случае, возникло исключение или нет. Он полезен для закрытия ресурсов (файлы, соединения) независимо от результата выполнения кода.

else: используется для выполнения кода, если в блоке try не возникло исключений. Это позволяет отделить основной рабочий код от кода, который должен выполняться только при успешном выполнении блока try.

4. Как осуществляется генерация исключений в языке Python?

Исключения в Python можно создавать вручную с помощью оператора raise, который генерирует исключение определенного типа и позволяет передать сообщение об ошибке.

5. Как создаются классы пользовательских исключений в языке Python?

Для создания пользовательских исключений создается класс, наследующий от базового класса исключений, например, Exception. Это позволяет задать специфичное поведение или сообщение для исключений в рамках программы.

6. Каково назначение модуля logging?

Модуль logging предназначен для регистрации (логгирования) событий, возникающих во время выполнения программы. Он позволяет записывать

сообщения об ошибках, предупреждениях, информацию о работе и т.п. в файл или консоль.

7. Какие уровни логгирования поддерживаются модулем logging? Приведите примеры, в которых могут быть использованы сообщения с этим уровнем журналирования.

Модуль logging поддерживает несколько уровней:

DEBUG: для отладки, используется для детализированной информации о ходе выполнения.

INFO: для записи общей информации, например, о начале и завершении работы.

WARNING: для потенциально опасных ситуаций, например, о приближении к лимитам использования ресурсов.

ERROR: для ошибок, которые не приводят к завершению работы программы, например, невозможность открыть файл.

CRITICAL: для ошибок, которые могут привести к аварийному завершению программы, например, потеря доступа к базе данных.

Вывод: Приобрел навыки по созданию иерархии классов при написании программ с помощью языка программирования Python версии 3.x.