

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

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО РАБОТЕ №2.15
дисциплины «Программирование на языке Python»

Выполнил:

Баканов Артем Вадимович
2 курс, группа ИТС-б-о-22-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи», направленность
(профиль)
«Инфокоммуникационные системы и
сети», очная форма обучения

(подпись)

Руководитель практики:

Воронкин Р.А., канд. тех. наук, доцент, доцент
кафедры инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

г.Ставрополь, 2023

Тема: работа с файлами на языке Python.

Цель работы: приобретение навыков по работе с текстовыми файлами при написании программ с помощью языка программирования Python версии 3.x, изучение основных методов модуля os для работы с файловой системой, получение аргументов командной строки.

Порядок выполнения работы:

Задание 1.

Изучил теоретический материал работы, создал общедоступный репозиторий на GitHub, в котором использована лицензий MIT и язык программирования Python, также добавил файл .gitignore с необходимыми правилами.

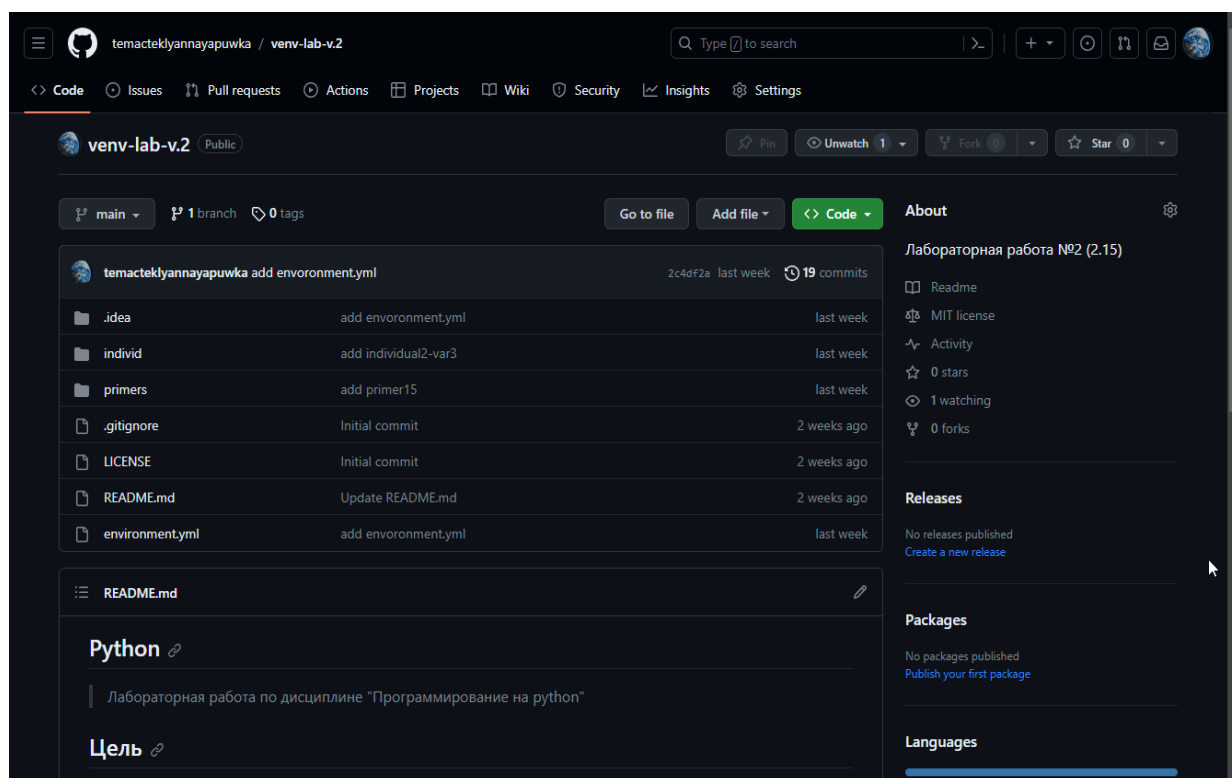


Рисунок 1. Новый репозиторий

Задание 2.

Проклонировал свой репозиторий на свой компьютер.

Организовал свой репозиторий в соответствии с моделью ветвления gitflow, появилась новая ветка develop.

```
C:\Users\HUAWEI>cd C:\Users\HUAWEI\Lab.Rab.2.15
C:\Users\HUAWEI\Lab.Rab.2.15>git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
C:\Users\HUAWEI\Lab.Rab.2.15>git init flow
Initialized empty Git repository in C:/Users/HUAWEI/Lab.Rab.2.15/flow/.git/
C:\Users\HUAWEI\Lab.Rab.2.15>git flow init
Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]
How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/HUAWEI/Lab.Rab.2.15/.git/hooks]
C:\Users\HUAWEI\Lab.Rab.2.15>git status
On branch develop
```

Рисунок 2. Клонирование и модель ветвления git-flow

Реализовывал примеры и индивидуальные задания на основе ветки develop, без создания дополнительной ветки feature/(название ветки) по указанию преподавателя.

Задание 3. Работа с примерами.

Работа с примером №1.

Условие примера: чтение первых двух строк с помощью метода readline().

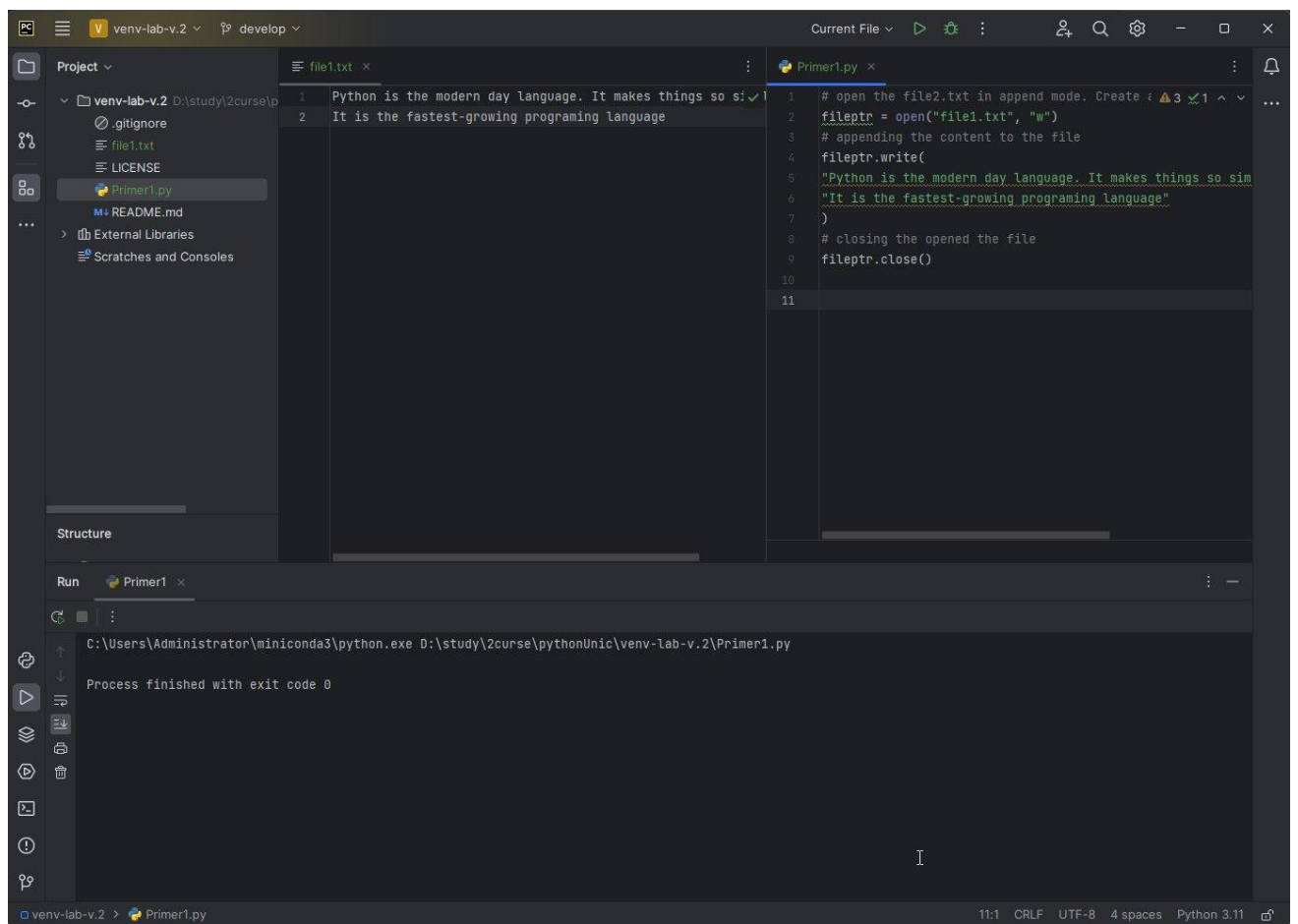


Рисунок 3. Пример с readline

Работа с примером №2.

Условие примера: написать программу, которая считывает текст из файла и выводит на экран только предложения, содержащие запятые. Каждое предложение в файле записано на отдельной строке.

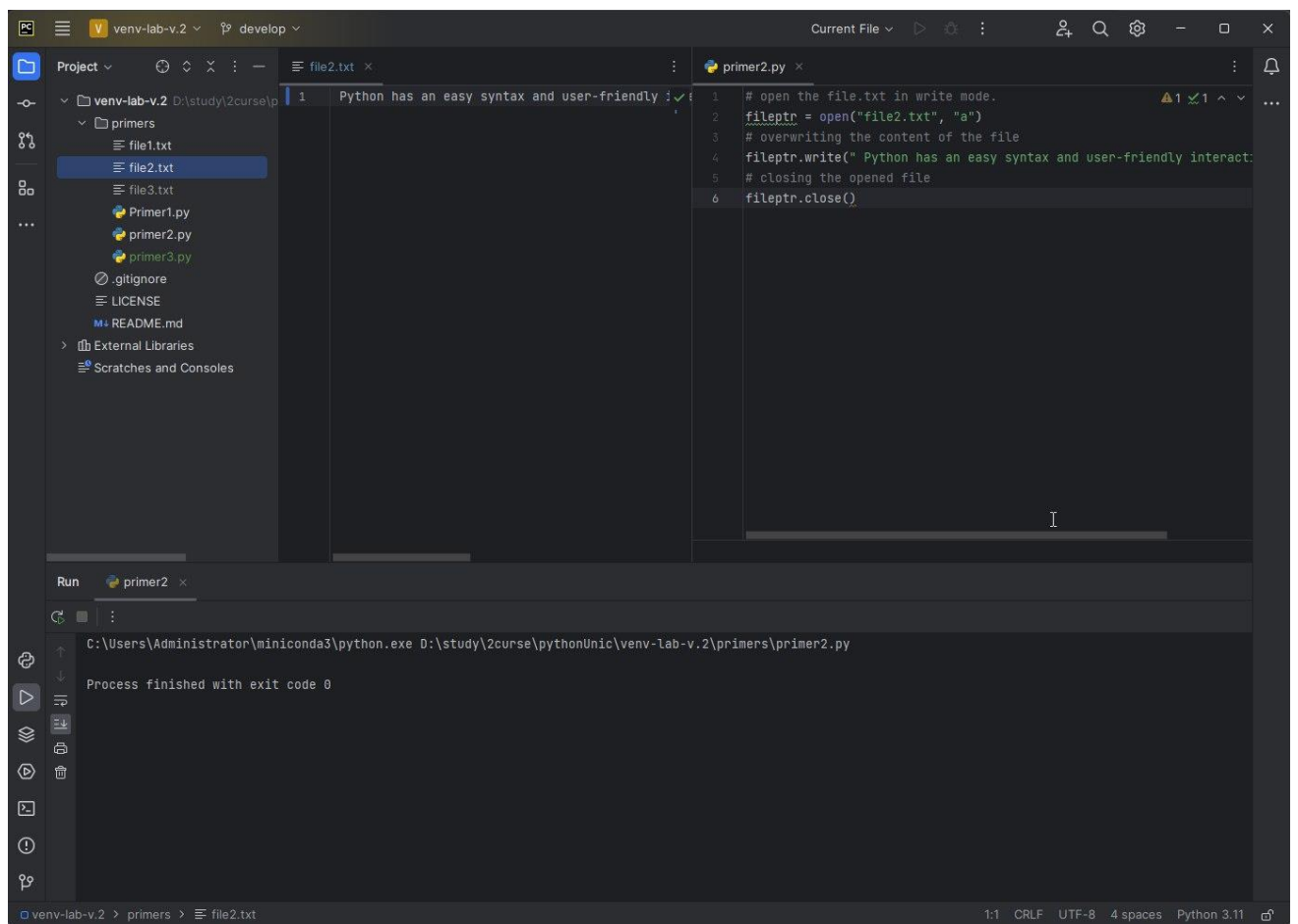


Рисунок 4. Пример с запятыми

Работа с примером №3.

Условие примера: python предоставляет метод `tell()`, который используется для печати номера байта, в котором в настоящее время существует указатель файла.

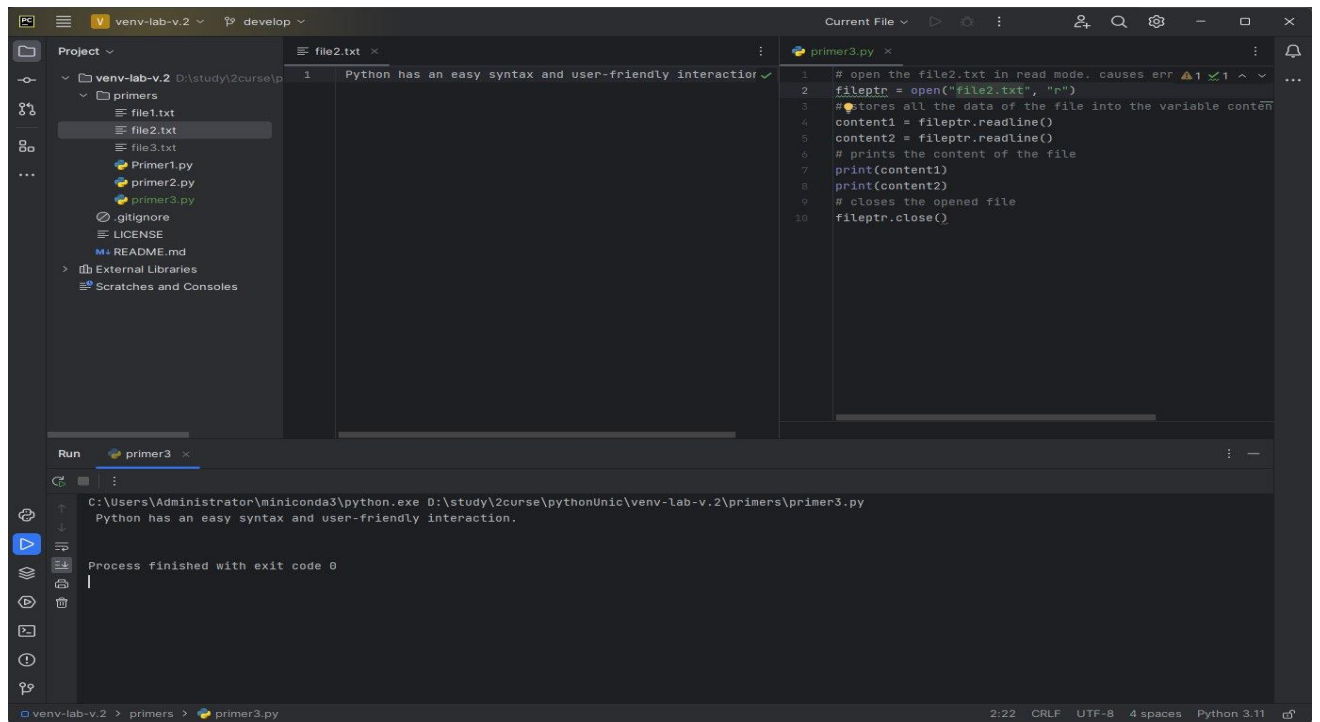


Рисунок 4. Пример печати номера байта

Задание 3

Выполнение индивидуального задания №1. (Вариант №3)

Написать программу, которая считывает текст из файла и выводит на экран только строки, содержащие двузначные числа.

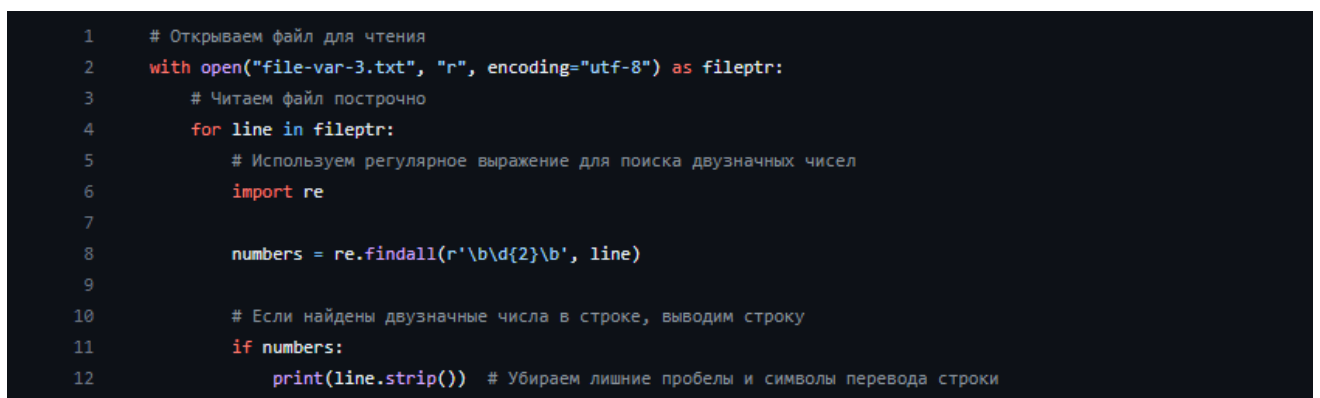


Рисунок 5. Индивидуальное задание 1

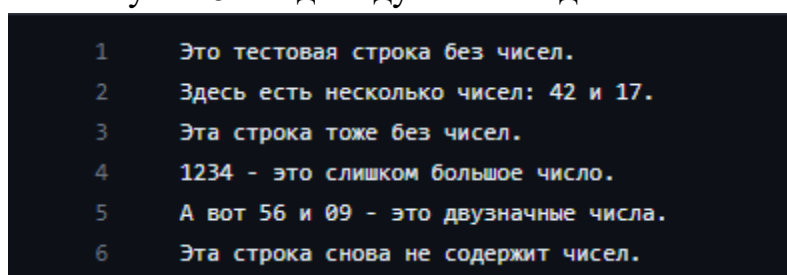
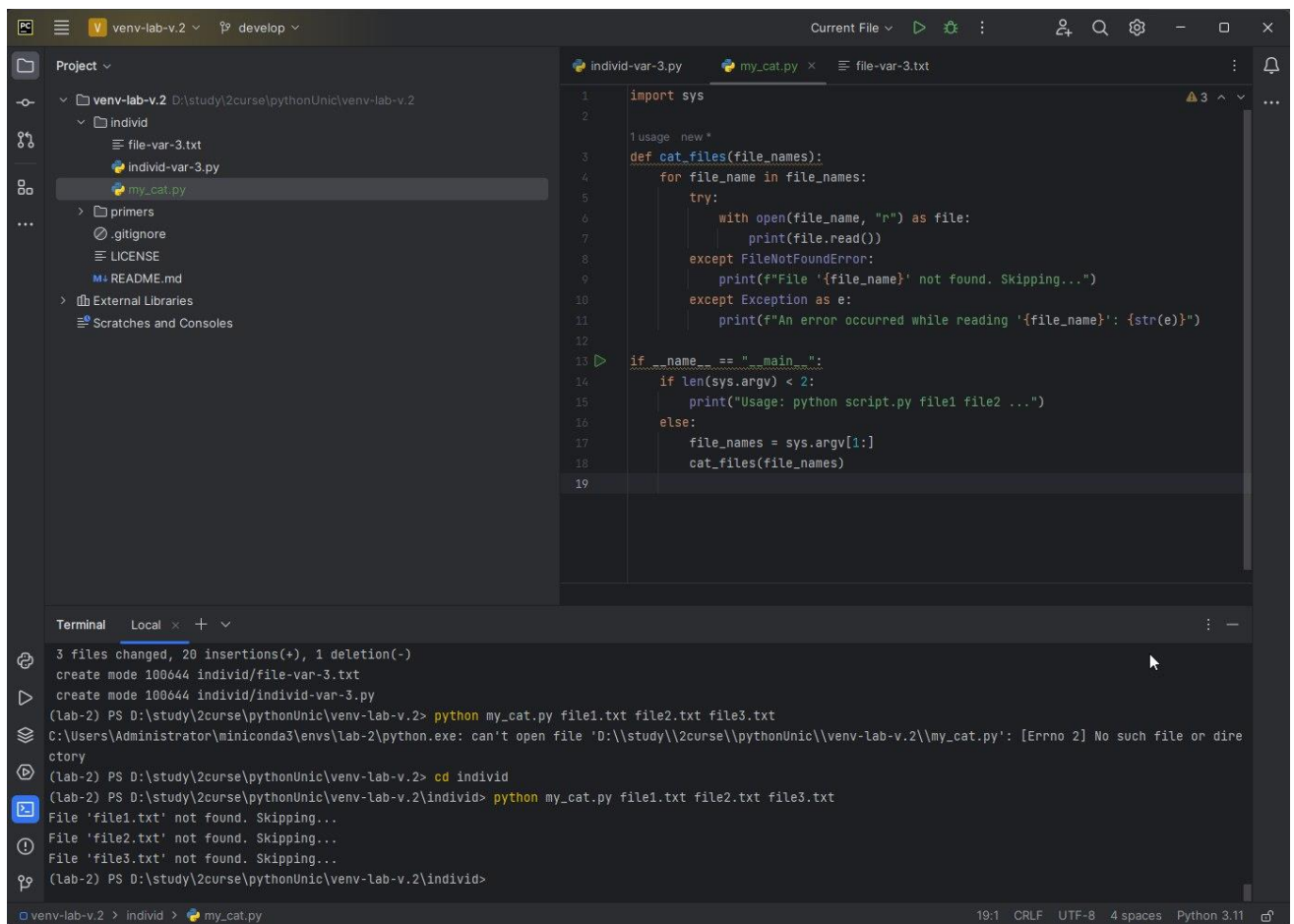


Рисунок 6. Результат задания 1

Задание 4

Выполнение индивидуального задания №2. (Вариант №3)

Продолжаем тему операционных систем на базе Unix, в которых обычно также есть утилита с названием `cat`, что является сокращением от `concatenate` (сцепить). Эта утилита выводит на экран объединенное содержимое нескольких файлов, имена которых передаются ей в качестве аргументов командной строки. При этом файлы сцепляются в том порядке, в котором указаны в аргументах. Напишите программу на Python, имитирующую работу этой утилиты. В процессе работы программа должна выдавать сообщения о том, какие файлы открыть не удастся, и переходить к следующим файлам. Если программа была запущена без аргументов командной строки, на экран должно быть выведено соответствующее сообщение об ошибке.



The screenshot shows an IDE with a project named 'venv-lab-v.2'. The file explorer on the left shows the project structure, including 'indiv-var-3.py' and 'my_cat.py'. The main editor window displays the code for 'my_cat.py', which implements a Python script that reads and concatenates the contents of multiple files. The script includes error handling for files that are not found or cannot be opened. The terminal at the bottom shows the execution of the script, including the creation of test files and the output of the script when run with and without arguments.

```
1 import sys
2
3 usage = """
4 def cat_files(file_names):
5     for file_name in file_names:
6         try:
7             with open(file_name, "r") as file:
8                 print(file.read())
9         except FileNotFoundError:
10            print(f"File '{file_name}' not found. Skipping...")
11        except Exception as e:
12            print(f"An error occurred while reading '{file_name}': {str(e)}")
13
14 if __name__ == "__main__":
15     if len(sys.argv) < 2:
16         print("Usage: python script.py file1 file2 ...")
17     else:
18         file_names = sys.argv[1:]
19         cat_files(file_names)
```

Terminal output:

```
3 files changed, 20 insertions(+), 1 deletion(-)
create mode 100644 indiv-var-3.py
create mode 100644 my_cat.py
(Lab-2) PS D:\study\2course\python\venv-lab-v.2> python my_cat.py file1.txt file2.txt file3.txt
C:\Users\Administrator\miniconda3\envs\lab-2\python.exe: can't open file 'D:\study\2course\python\venv-lab-v.2\my_cat.py': [Errno 2] No such file or directory
(Lab-2) PS D:\study\2course\python\venv-lab-v.2> cd indiv-var-3
(Lab-2) PS D:\study\2course\python\venv-lab-v.2\indiv-var-3> python my_cat.py file1.txt file2.txt file3.txt
File 'file1.txt' not found. Skipping...
File 'file2.txt' not found. Skipping...
File 'file3.txt' not found. Skipping...
(Lab-2) PS D:\study\2course\python\venv-lab-v.2\indiv-var-3>
```

Рисунок 7. Выполнение индивидуального задания 2

Ссылка на репозиторий: <https://github.com/temacteklyannayapuwka/venv-lab-v.2>

Контрольные вопросы:

1. Как открыть файл в языке Python только для чтения?

Для открытия файла в режиме только для чтения в Python используется функция `open` с аргументом `'r'`.

2. Как открыть файл в языке Python только для записи?

Для открытия файла в режиме только для записи в Python используется функция `open` с аргументом `'w'`.

3. Как прочитать данные из файла в языке Python?

Чтение данных из файла в Python можно осуществить с помощью метода `read`.

4. Как записать данные в файл в языке Python?

Запись данных в файл в Python можно осуществить с помощью метода `write`.

5. Как закрыть файл в языке Python?

Чтобы закрыть файл в Python после работы с ним, используется метод `close`.

6. Изучите самостоятельно работу конструкции `with ... as`. Каково ее назначение в языке Python? Где она может быть использована еще, помимо работы с файлами?

Конструкция `with ... as` используется в Python для автоматического управления ресурсами. В случае работы с файлами, она гарантирует, что файл будет автоматически закрыт по окончании блока кода, независимо от того, произошло исключение или нет.

7. Изучите самостоятельно документацию Python по работе с файлами. Какие помимо рассмотренных существуют методы записи/чтения информации из файла?

Помимо методов `read()` и `write()`, существуют другие методы работы с файлами в Python, такие как `readline()` для чтения одной строки из файла, `readlines()` для чтения всех строк из файла в список, `seek()` для перемещения указателя чтения/записи по файлу, `tell()` для получения текущей позиции указателя и другие. Подробнее о методах работы с файлами можно узнать из документации Python.

8. Какие существуют, помимо рассмотренных, функции модуля `os` для работы с файловой системой?

Некоторые функции модуля `os` для работы с файловой системой в Python:

- `os.rename()` для переименования файла или директории.
- `os.remove()` для удаления файла.
- `os.mkdir()` для создания директории.
- `os.getcwd()` для получения текущей рабочей директории.
- `os.path.exists()` для проверки существования файла или директории.
- `os.path.isfile()` для проверки, является ли путь файлом.
- `os.path.isdir()` для проверки, является ли путь директорией.
- `os.path.join()` для объединения путей к файлам или директориям.

Вывод: приобрел навыки по работе с текстовыми файлами при написании программ с помощью языка программирования Python версии 3.x, изучили основных методов модуля `os` для работы с файловой системой, получение аргументов командной строки.