

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 ЛАБОРАТОРНОЙ РАБОТЕ №2.15**  
**дисциплины «Основы программной инженерии»**

Выполнил:  
Баратов Семен Григорьевич  
2 курс, группа ПИЖ-б-о-22-1,  
09.03.04 «Программная инженерия»,  
направленность (профиль) «Разработка  
и сопровождение программного  
обеспечения», очная форма обучения

---

(подпись)

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

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2024 г.

## Тема: Работа с файлами в языке Python

**Цель:** 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.

### Результаты выполнения

1. Создали репозиторий с лицензией MIT, добавили в .gitignore необходимые правила для работы с IDE PyCharm, клонировали репозиторий, организовали репозиторий в соответствии с моделью git-flow.

```
Last login: Tue Oct 24 20:29:02 on ttys000
itssyoma@MacBook-Air-Sema Основы программной инженерии % git clone https://github.com/itssyoma/megarepo_21.git
Cloning into 'megarepo_21'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
itssyoma@MacBook-Air-Sema Основы программной инженерии % cd megarepo_21
itssyoma@MacBook-Air-Sema megarepo_21 % git checkout -b develop
Switched to a new branch 'develop'
itssyoma@MacBook-Air-Sema megarepo_21 % git branch release
itssyoma@MacBook-Air-Sema megarepo_21 % git branch develop
fatal: a branch named 'develop' already exists
itssyoma@MacBook-Air-Sema megarepo_21 % git branch hotfix
itssyoma@MacBook-Air-Sema megarepo_21 % git branch feature
itssyoma@MacBook-Air-Sema megarepo_21 %
```

Рисунок 1 – Работа с репозиторием в командной строке.

2. Проработали примеры лабораторной работы.



Рисунок 2 – Примеры лабораторной работы.

3. Выполнили индивидуальное задание №1. Написать программу, которая считывает из текстового файла три предложения и выводит их в обратном порядке.

```
individual > ind1.py > ...
1 # /usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 # Написать программу, которая считывает из текстового
5 # файла три предложения и выводит их в обратном порядке.
6
7
8 if __name__ == "__main__":
9     with open('individual/ind1.txt', 'r') as file:
10         content = file.readlines()
11
12     for line in reversed(content):
13         print(line.strip())
14
```

ПРОБЛЕМЫ Выходные данные Консоль отладки **ТЕРМИНАЛ** Г

```
• (env) itssyoma@MacBook-Air-Sema megarepo_215 % "/Users/itssyoma/Ya
megarepo_215/env/bin/python" "/Users/itssyoma/Yandex.Disk.localize
idual/ind1.py"
3. Это третья строка
2. Это вторая строка
1. Это первая строка
○ (env) itssyoma@MacBook-Air-Sema megarepo_215 %
```

Рисунок 3 – Индивидуальное задание №1.

4. Выполнили индивидуальное задание №2. В операционных системах на базе Unix обычно присутствует утилита с названием head. Она выводит первые десять строк содержимого файла, имя которого передается в качестве аргумента командной строки. Напишите программу на Python, имитирующую поведение этой утилиты. Если файла, указанного пользователем, не существует, или не задан аргумент командной строки, необходимо вывести соответствующее сообщение об ошибке.

```
individual > ind2.py > ...
1 # /usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 # В операционных системах на базе Unix обычно присутствует утилита
5 # с названием head. Она выводит первые десять строк содержимого файла,
6 # имя которого передается в качестве аргумента командной строки.
7 # Напишите программу на Python, имитирующую поведение этой утилиты.
8 # Если файла, указанного пользователем, не существует, или не задан
9 # аргумент командной строки, необходимо вывести соответствующее сообщение об ошибке.
10
11 import sys
12 import os
13
14 def check_file_exists(file_path):
15     return os.path.exists(file_path)
16
17 if __name__ == "__main__":
18     if len(sys.argv) < 2:
19         print("Add file path as an argument", file=sys.stderr)
20         sys.exit(1)
21
22     file_path = sys.argv[1]
23
24     if not check_file_exists(file_path):
25         print("No such file or directory", file=sys.stderr)
26         sys.exit(1)
27
28     with open(file_path, 'r') as file:
29         i = 0
30         while i != 10:
31             line = file.readline()
32             print(line.strip())
33             i += 1
34
```


ПРОБЛЕМЫ Выходные данные Консоль отладки **ТЕРМИНАЛ** ПОРТЫ

```
• (env) itssyoma@MacBook-Air-Sema megarepo_215 % python3 individual/ind2.py
Add file path as an argument
• (env) itssyoma@MacBook-Air-Sema megarepo_215 % python3 individual/ind2.py text.txt
No such file or directory
• (env) itssyoma@MacBook-Air-Sema megarepo_215 % python3 individual/ind2.py individual/ind2.txt
1. Солнце светит ярко на небе.
2. Коты любят играть с мячиками.
3. Ветер шумит в листве деревьев.
4. Дети с удовольствием катаются на карусели.
5. Море шумит и пенится у берега.
6. Птицы радостно щебечут на ветках деревьев.
7. Цветы расцветают после дождя.
8. Луна светит ярким светом в ночном небе.
9. Друзья смеются и разговаривают за чашкой кофе.
10. Звезды мерцают на темном небе.
```

Рисунок 4 – Индивидуальное задание №2.

5. Самостоятельно придумали и решили задачу с использованием модуля `os` – Напишите программу, которая будет выводить список всех файлов в указанной директории и их размеры в байтах.

```
individual > ind3.py > ...
1  # /usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  # Напишите программу, которая будет выводить список
5  # всех файлов в указанной директории и их размеры в байтах.
6
7
8  import os
9  import sys
10
11
12  if __name__ == "__main__":
13      directory = sys.argv[1]
14      files = os.listdir(directory)
15      for file in files:
16          file_path = os.path.join(directory, file)
17          if os.path.isfile(file_path):
18              size = os.path.getsize(file_path)
19              print(f"File: {file} | Size: {size} bytes")
20
```



ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ    ПОРТЫ

```
• (env) itssyoma@MacBook-Air-Sema megarepo_215 % python3 individual/ind3.py "/"
новы программной инженерии/megarepo_215"
File: .flake8 | Size: 83 bytes
File: .DS_Store | Size: 6148 bytes
File: LICENSE | Size: 1070 bytes
File: requirements.txt | Size: 59 bytes
File: .pre-commit-config.yaml | Size: 464 bytes
File: pyproject.toml | Size: 276 bytes
File: README.md | Size: 91 bytes
File: .gitignore | Size: 3101 bytes
```

Рисунок 5 – Индивидуальное задание №3.

### Ответы на контрольные вопросы

1. Для открытия файла только для чтения в Python используется функция `open()` с параметром `'r'`. Например:

```
file = open('file.txt', 'r')
```

2. Для открытия файла только для записи в Python используется функция `open()` с параметром `'w'`. Например:

```
file = open('file.txt', 'w')
```

3. Для чтения данных из файла в Python можно использовать метод `read()`. Например:

```
data = file.read()
```

4. Для записи данных в файл в Python можно использовать метод `write()`. Например:

```
file.write("Hello, world!")
```

5. Чтобы закрыть файл в Python, используйте метод `close()`. Например:

```
file.close()
```

6. Конструкция `with ... as` используется для автоматического управления ресурсами, таких как файлы. Она гарантирует, что ресурсы будут правильно освобождены даже в случае возникновения исключений. Кроме работы с файлами, она может быть использована, например, для работы с сетевыми соединениями или базами данных.

7. Помимо методов `read()` и `write()`, существуют такие методы работы с файлами как `readline()`, `readlines()`, `writelines()` и другие. Эти методы позволяют более гибко читать и записывать данные из/в файл.

8. Модуль `os` предоставляет различные функции для работы с файловой системой. Некоторые из них включают `os.rename()`, `os.remove()`, `os.mkdir()`, `os.rmdir()` и другие, позволяющие перемещать, удалять и создавать файлы и директории.