

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 №22**  
**дисциплины «Основы программной инженерии»**

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

\_\_\_\_\_  
(подпись)

Проверил Воронкин Роман Александрович

\_\_\_\_\_  
(подпись)

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

Ставрополь, 2024 г.

**Тема:** Работа с файловой системой в Python3 с использованием модуля pathlib.

**Цель работы:** Приобретение навыков по работе с файловой системой с помощью библиотеки pathlib языка программирования Python версии 3.x.

Ход работы.

### 1. Создание нового репозитория с лицензией MIT.

**Create a new repository**

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

---

*Required fields are marked with an asterisk (\*).*

**Owner \*** SashkaHacker / **Repository name \*** laba10

✔ laba10 is available.

Great repository names are short and memorable. Need inspiration? How about [fluffy-guide](#) ?

**Description** (optional)

---

☒ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

---

**Initialize this repository with:**

☐ **Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

**Add .gitignore**

.gitignore template: Python

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

**Choose a license**

License: MIT License

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

---

ⓘ You are creating a public repository in your personal account.

[Create repository](#)

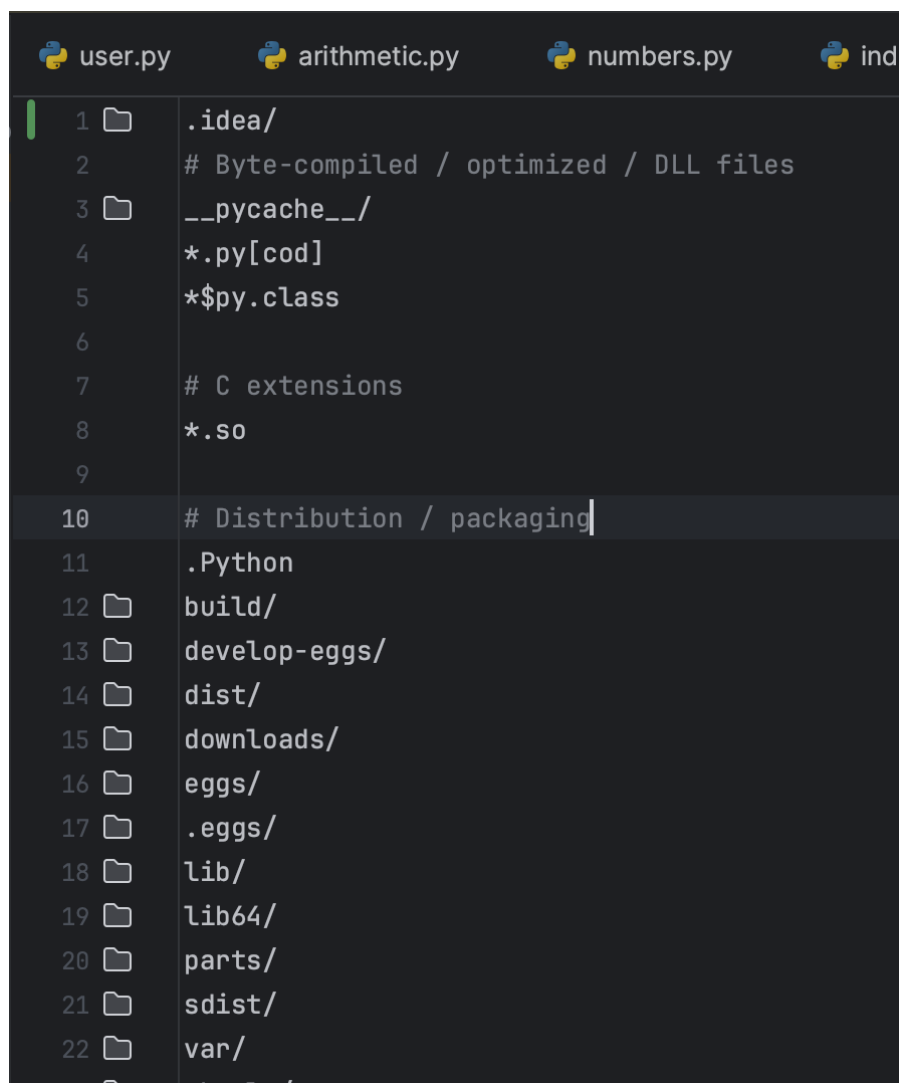
Рисунок 1 – Создание репозитория

2. Клонировал репозиторий на рабочий ПК.

```
[→ GitHub git clone https://github.com/SashkaHacker/laba16.git
Cloning into 'laba16'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
```

Рисунок 2 – Клонирование репозитория

3. Дополнил файл .gitignore необходимыми инструкциями.

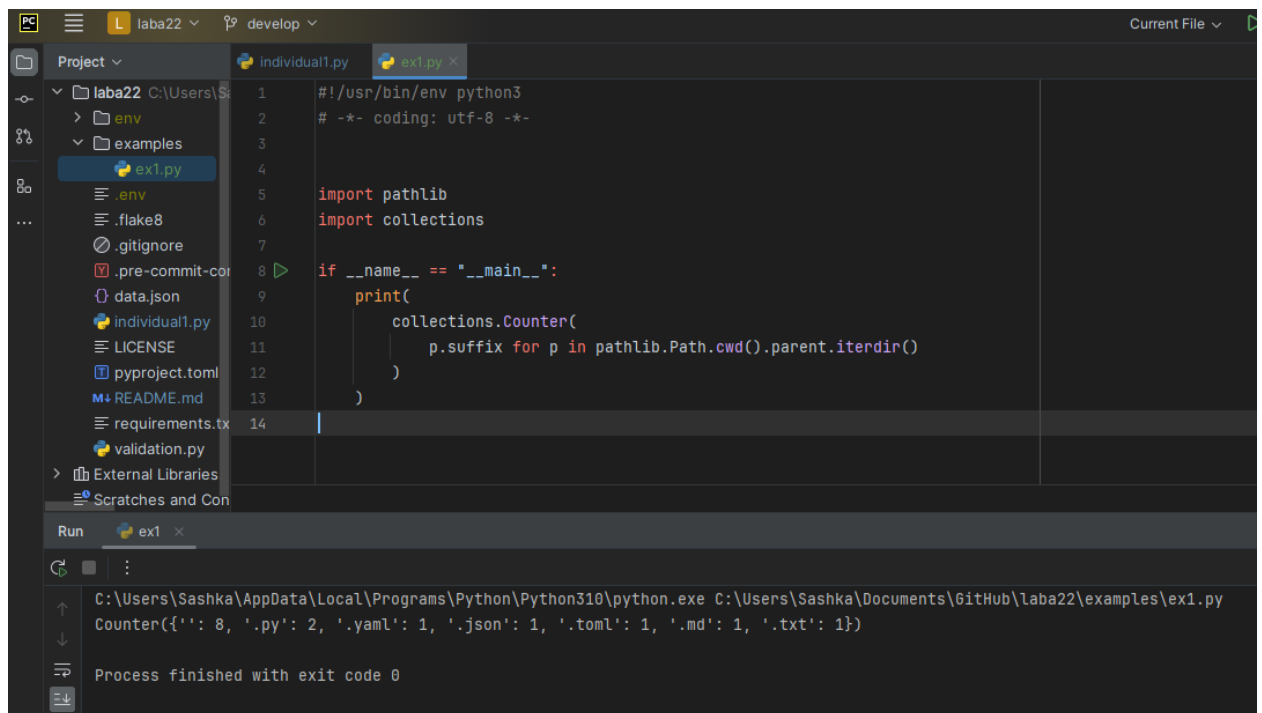


The image shows a code editor with a dark theme. At the top, there are four tabs: 'user.py', 'arithmetic.py', 'numbers.py', and 'ind'. The active tab is 'user.py'. The editor displays the content of a '.gitignore' file. The file contains the following text:

```
1  .idea/
2  # Byte-compiled / optimized / DLL files
3  __pycache__/
4  *.py[cod]
5  *$py.class
6
7  # C extensions
8  *.so
9
10 # Distribution / packaging
11 .Python
12 build/
13 develop-eggs/
14 dist/
15 downloads/
16 eggs/
17 .eggs/
18 lib/
19 lib64/
20 parts/
21 sdist/
22 var/
```

Рисунок 3 – Файл .gitignore

#### 4. Проработка примеров из лабораторной работы.

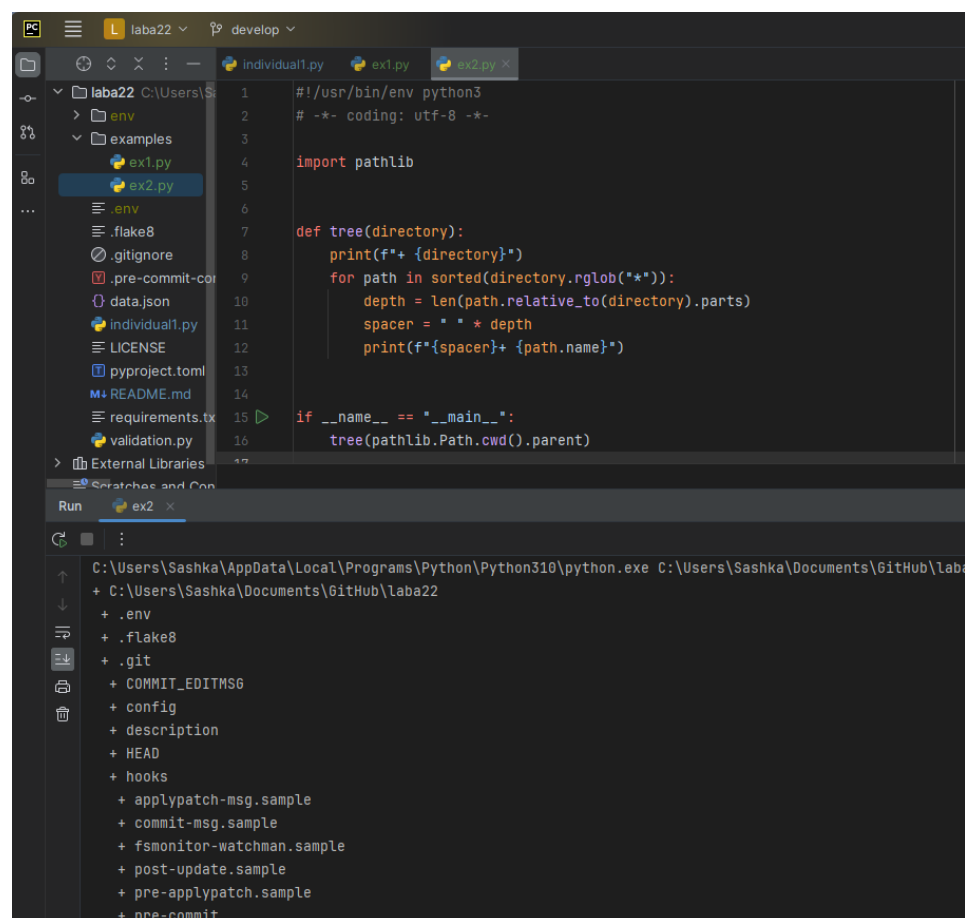


The screenshot shows a code editor with a project named 'laba22'. The file explorer on the left shows a directory structure with files like 'env', 'examples', 'ex1.py', '.env', '.flake8', '.gitignore', '.pre-commit-config.yaml', 'data.json', 'individual1.py', 'LICENSE', 'pyproject.toml', 'README.md', 'requirements.txt', and 'validation.py'. The main editor window displays the content of 'ex1.py'.

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 import pathlib
6 import collections
7
8 if __name__ == "__main__":
9     print(
10         collections.Counter(
11             p.suffix for p in pathlib.Path.cwd().parent.iterdir()
12         )
13     )
14
```

The Run window at the bottom shows the command executed: `C:\Users\Sashka\AppData\Local\Programs\Python\Python310\python.exe C:\Users\Sashka\Documents\GitHub\laba22\examples\ex1.py` and the output: `Counter({'': 8, '.py': 2, '.yaml': 1, '.json': 1, '.toml': 1, '.md': 1, '.txt': 1})`. The process finished with exit code 0.

Рисунок 4 – Демонстрация работы примера №1



The screenshot shows the same code editor with the file explorer displaying the same directory structure. The main editor window displays the content of 'ex2.py'.

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import pathlib
5
6
7 def tree(directory):
8     print(f"+ {directory}")
9     for path in sorted(directory.rglob("*")):
10         depth = len(path.relative_to(directory).parts)
11         spacer = "  " * depth
12         print(f"{spacer}+ {path.name}")
13
14
15 if __name__ == "__main__":
16     tree(pathlib.Path.cwd().parent)
17
```

The Run window at the bottom shows the command executed: `C:\Users\Sashka\AppData\Local\Programs\Python\Python310\python.exe C:\Users\Sashka\Documents\GitHub\laba22\examples\ex2.py`. The output is a tree structure of the directory:

```
+ C:\Users\Sashka\Documents\GitHub\laba22
+ .env
+ .flake8
+ .git
+ COMMIT_EDITMSG
+ config
+ description
+ HEAD
+ hooks
+ applypatch-msg.sample
+ commit-msg.sample
+ fsmonitor-watchman.sample
+ post-update.sample
+ pre-applypatch.sample
+ pre-commit
```

Рисунок 5 – Демонстрация работы примера №2

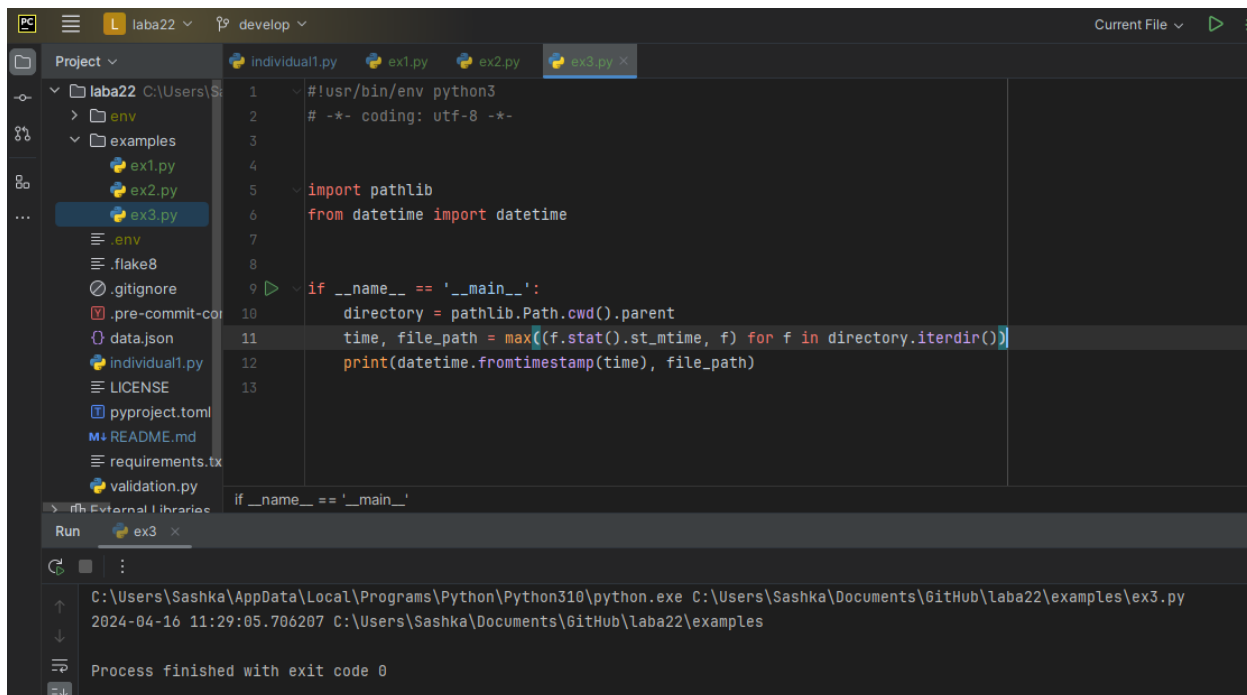


Рисунок 6 – Демонстрация работы примера №3

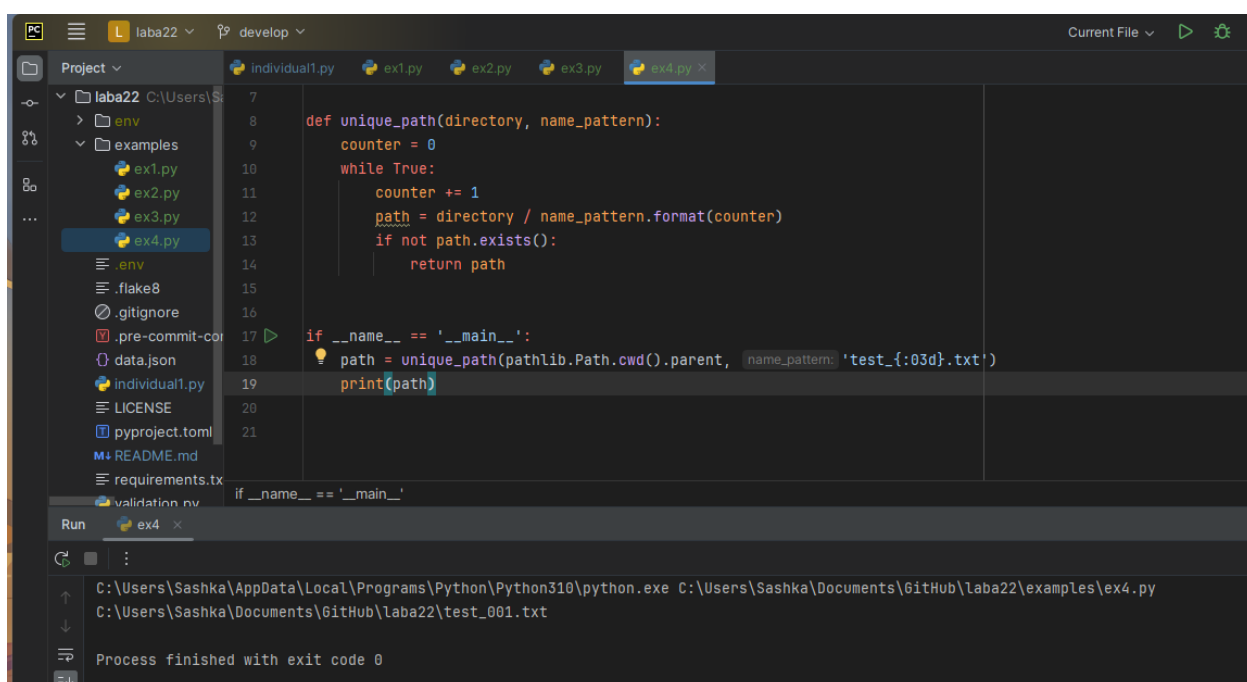


Рисунок 7 – Демонстрация работы примера №4

5. Выполнение индивидуального задания №1. Для своего варианта лабораторной работы 2.17 добавьте возможность хранения файла данных в

домашнем каталоге пользователя. Для выполнения операций с файлами необходимо использовать модуль pathlib.

```
(venv) → laba22 git:(develop) × python3 individual.py display data.json
```

№	Фамилия	Имя	Номер телефона	Дата рождения
1	Васильев	Василий	89034567890	30:08:1965
2	Сидоров	Сидор	89033456789	23:03:1975
3	Алексеев	Алексей	89035678901	17:11:1980
4	Иванов	Иван	89031234567	01:01:1985
5	Петров	Петр	89032345678	15:05:1990

```
(venv) → laba22 git:(develop) ×
```

Рисунок 8 – Демонстрация работы индивидуального задания №1

6. Выполнение индивидуального задания №2. Разработайте аналог утилиты tree в linux. Используйте возможности модуля argparse для управления отображением дерева каталогов файловой системы. Добавьте дополнительные уникальные возможности в данный программный продукт.

```
(venv) → laba22 git:(develop) × python3 individual2.py /Users --level=3
```

```
Users
├── aleksandrmatveev
│   ├── .config
│   ├── Music
│   ├── .docker
│   ├── Yandex.Disk.localized
│   ├── .local
│   ├── jan
│   ├── Pictures
│   ├── .ipython
│   ├── Desktop
│   ├── Library
│   ├── .oracle_jre_usage
│   ├── .android
│   ├── .pgadmin
│   ├── .cups
│   ├── PycharmProjects
│   ├── Public
│   ├── .yandex
│   ├── .idlerc
│   ├── .ssh
│   ├── Movies
│   └── Applications
```

Рисунок 9 – Демонстрация работы индивидуального задания №2

### Контрольные вопросы:

1. До Python 3.4 для работы с файловой системой использовались модули `os`, `os.path` и `shutil`.
2. PEP 428 регламентирует добавление модуля `pathlib` в стандартную библиотеку Python для предоставления объектно-ориентированного интерфейса для работы с путями файловой системы.
3. Создание путей в `pathlib` осуществляется путем создания экземпляра класса `Path`, например `Path('путь/к/каталогу')`.
4. Получить путь дочернего элемента можно используя оператор деления на объект `Path`, например `parent_path / 'child'`.
5. Путь к родительскому элементу можно получить через атрибут `parent` объекта `Path`, например `path.parent`.
6. Операции с файлами в `pathlib` выполняются с помощью методов объектов `Path`, таких как `open`, `read_text`, `write_text`, `unlink`.
7. Выделить компоненты пути можно с помощью атрибутов объекта `Path`, таких как `parts`, `name`, `suffix`, `stem`.
8. Для перемещения файлов используется метод `rename`, для удаления - метод `unlink` объектов `Path`.
9. Подсчет файлов осуществляется итерацией по объекту `Path` с использованием метода `iterdir` и проверкой `is_file`.
10. Для отображения дерева каталогов используется рекурсивный обход с помощью методов `iterdir` и `is_dir` и вывод информации в консоль.
11. Уникальное имя файла можно создать с помощью модуля `uuid` или проверкой на существование файла перед его созданием.
12. Отличия использования `pathlib` для разных операционных систем в основном касаются синтаксиса путей (например, использование обратного следа в Windows) и системных каталогов, но сам модуль абстрагирует эти различия и предоставляет единообразный интерфейс.