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 №18
**дисциплины «Основы программной инженерии»**

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

_____
(подпись)

Руководитель практики:
Воронкин Р. А., доцент кафедры
инфокоммуникаций

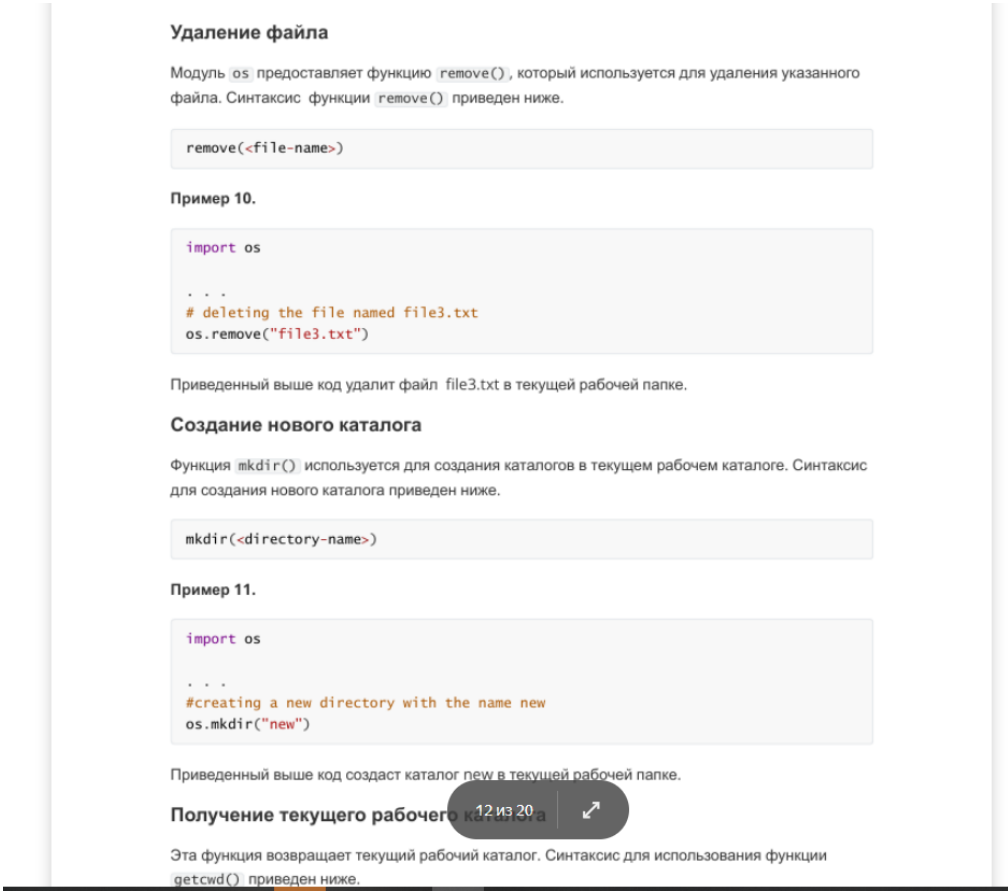
_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2023 г.

## Ход работы

1. Изучаем теоретический материал работы



### Удаление файла

Модуль `os` предоставляет функцию `remove()`, который используется для удаления указанного файла. Синтаксис функции `remove()` приведен ниже.

```
remove(<file-name>)
```

**Пример 10.**

```
import os

. . .
# deleting the file named file3.txt
os.remove("file3.txt")
```

Приведенный выше код удалит файл file3.txt в текущей рабочей папке.

### Создание нового каталога

Функция `mkdir()` используется для создания каталогов в текущем рабочем каталоге. Синтаксис для создания нового каталога приведен ниже.

```
mkdir(<directory-name>)
```

**Пример 11.**

```
import os

. . .
#creating a new directory with the name new
os.mkdir("new")
```

Приведенный выше код создаст каталог new в текущей рабочей папке.

12 из 20

### Получение текущего рабочего каталога

Эта функция возвращает текущий рабочий каталог. Синтаксис для использования функции `getcwd()` приведен ниже.

Рисунок 1.1 – Изучение материала для лабораторной работы

2. Создаем общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python

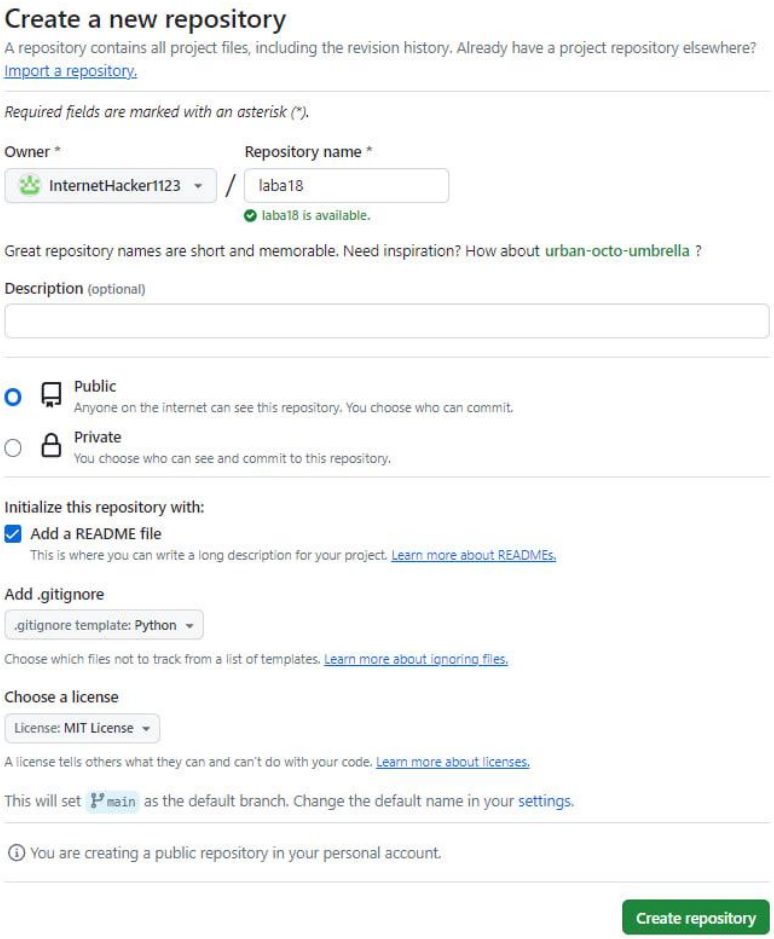



Рисунок 2.1 – Настройка репозитория

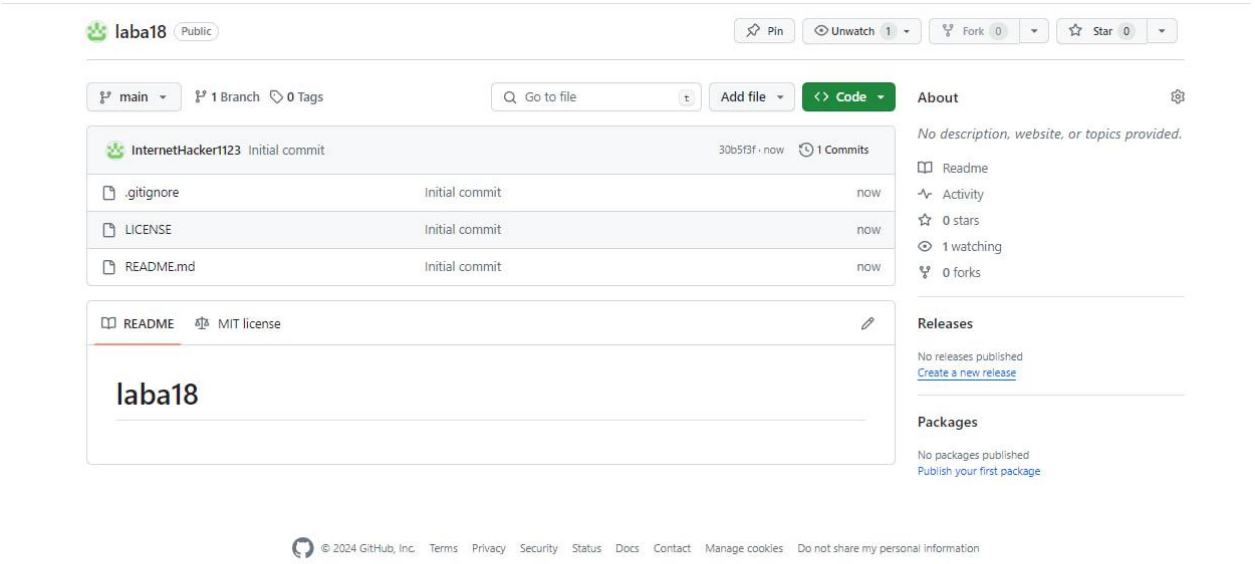



Рисунок 2.2 – Готовый репозиторий

3. Выполняем клонирование созданного репозитория

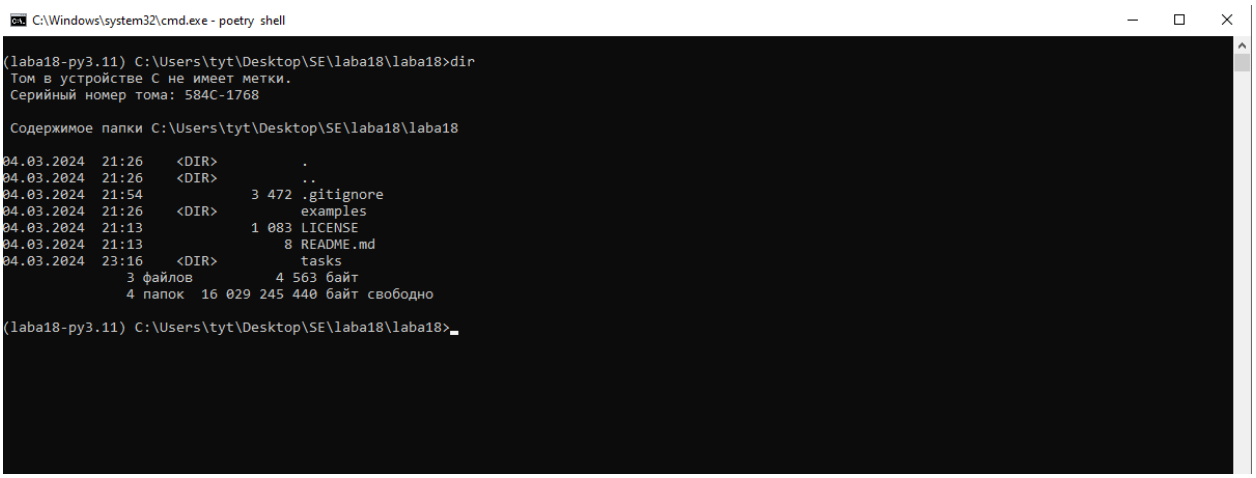



Рисунок 3.1 – Клонирование репозитория на локальный диск

4. Дополняем файл .gitignore необходимыми правилами для работы с IDE PyCharm

```
149  # pytype static type analyzer
150  .pytype/
151
152  # Cython debug symbols
153  cython_debug/
154
155  .vscode/*
156  !.vscode/settings.json
157  !.vscode/tasks.json
158  !.vscode/launch.json
159  !.vscode/extensions.json
160  !.vscode/*.code-snippets
161
162  # Local History for Visual Studio Code
163  .history/
164
165  # Built Visual Studio Code Extensions
166  *.vsix
167
168  # PyCharm
```

Рисунок 4.1 – .gitignore для VSCode

5. Организовываем свой репозиторий в соответствии с моделью ветвления git-flow

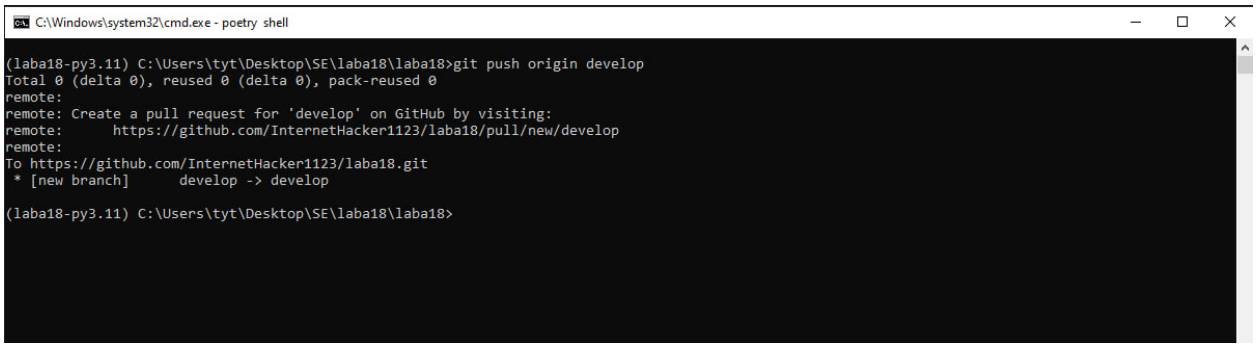



Рисунок 5.1 – Создание ветки develop от ветки main

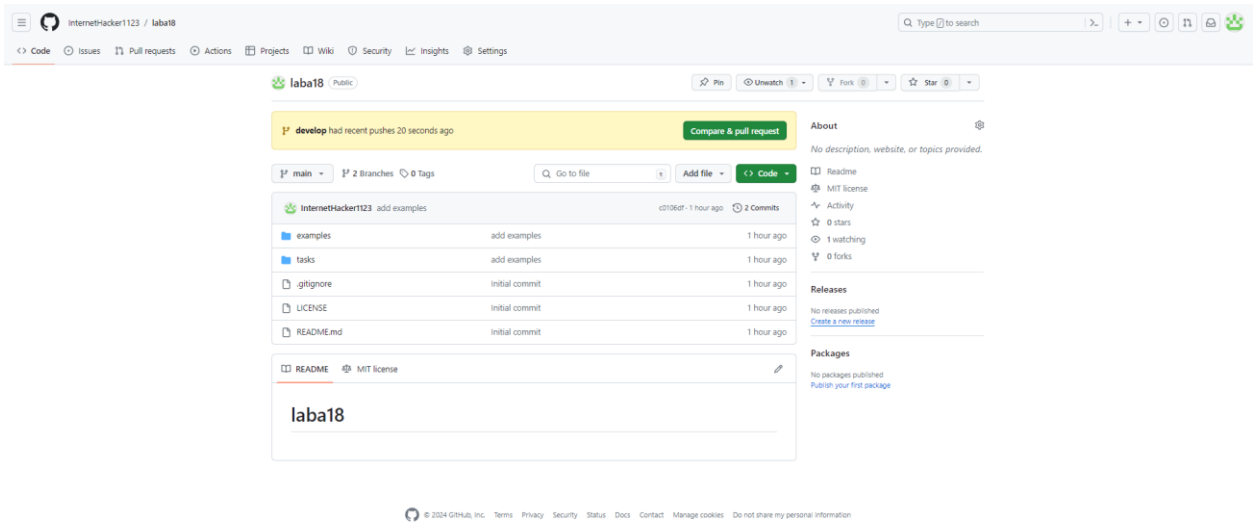


Рисунок 5.2 – Ветка develop на GitHub

6. Создаем проект PyCharm в папке репозитория

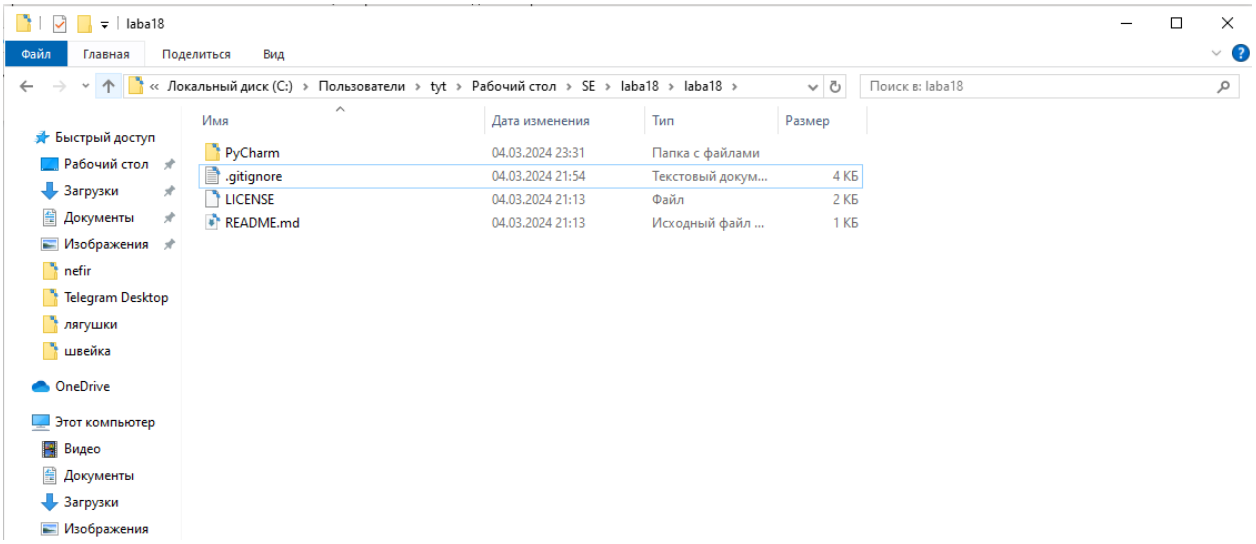



Рисунок 6.1 – Создание проекта PyCharm

7. Проработаем примеры лабораторной работы

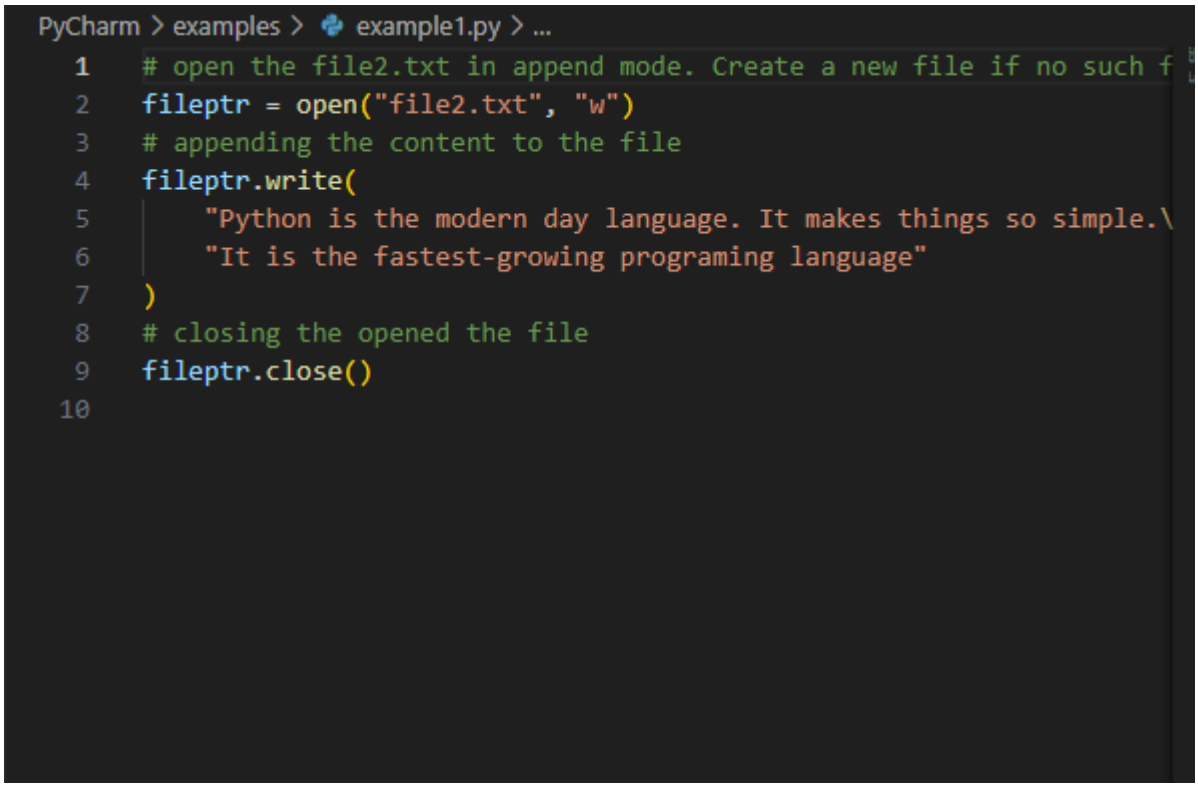



Рисунок 7.1 – Пример №1

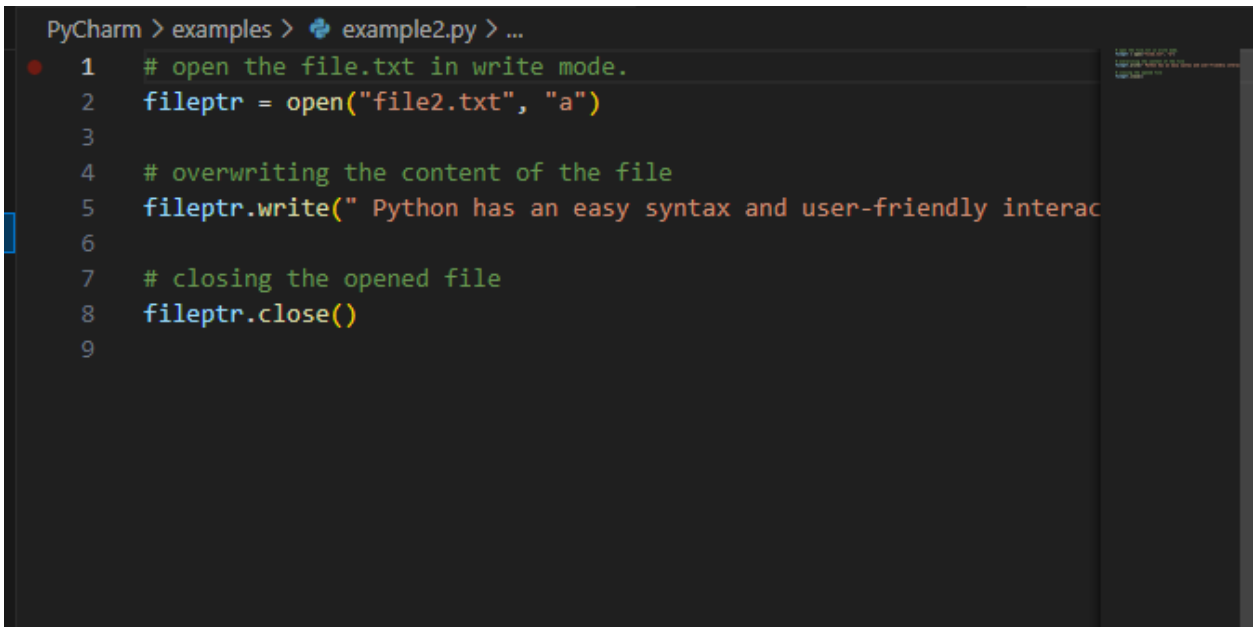



Рисунок 7.2 – Пример №2

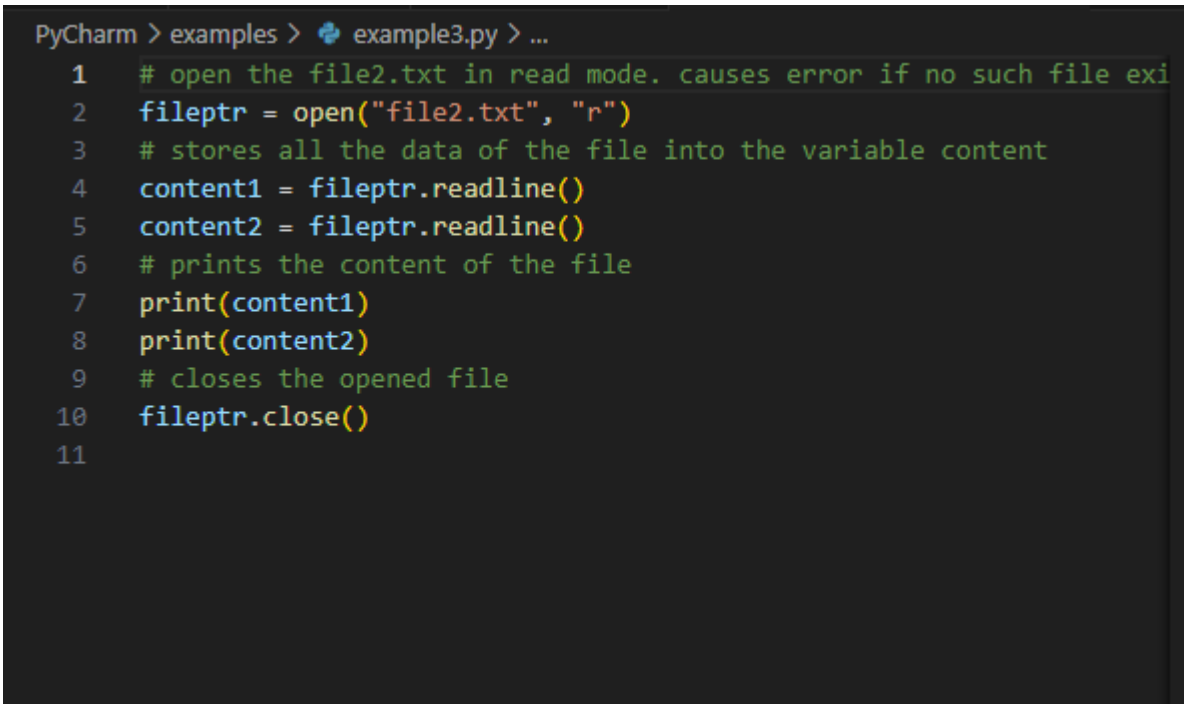



Рисунок 7.3 – Пример №3

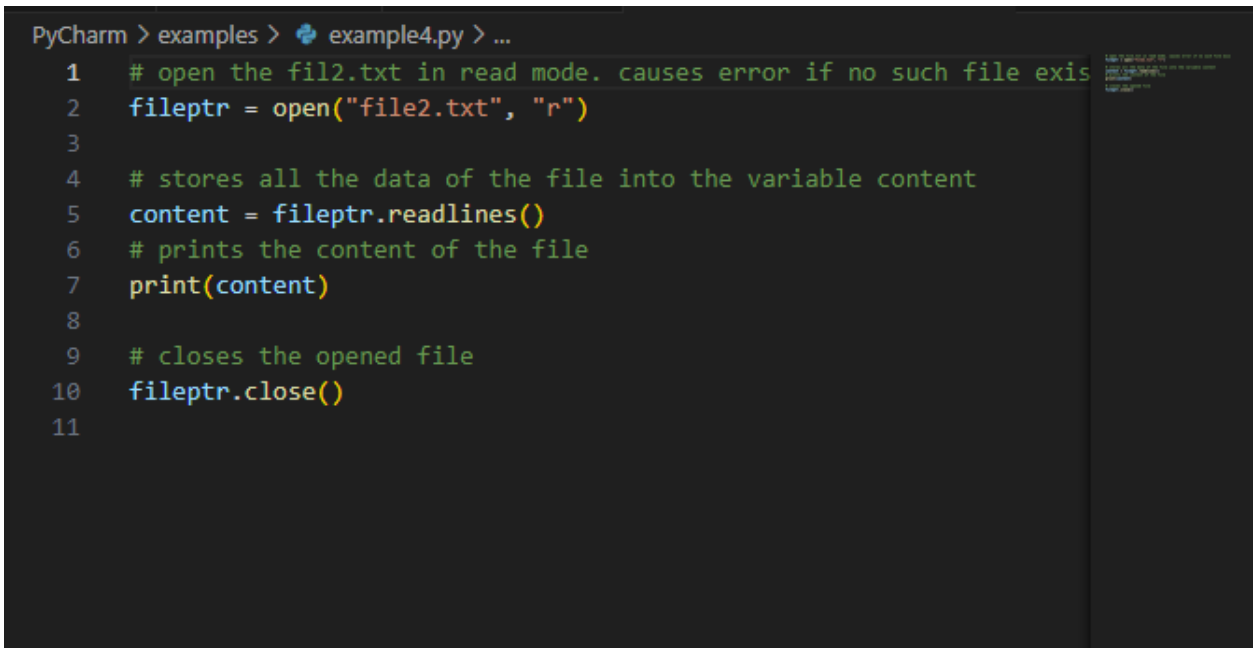



Рисунок 7.4 – Пример №4

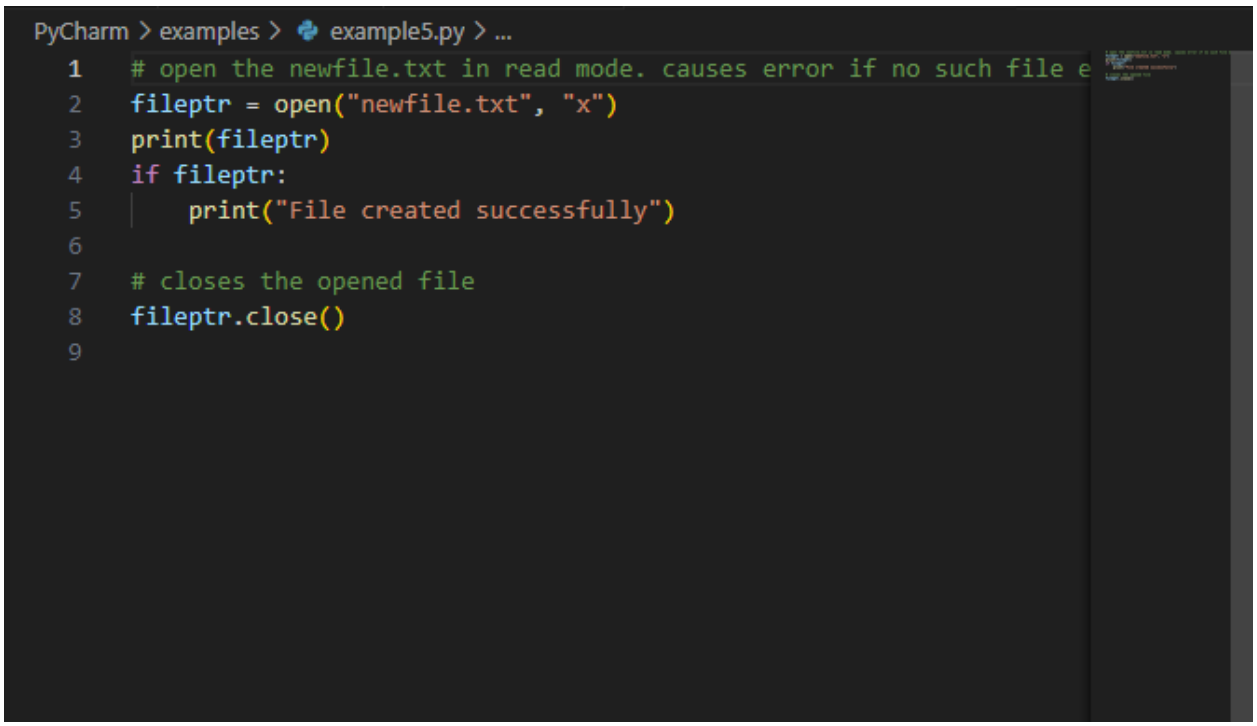



Рисунок 7.5 – Пример №5

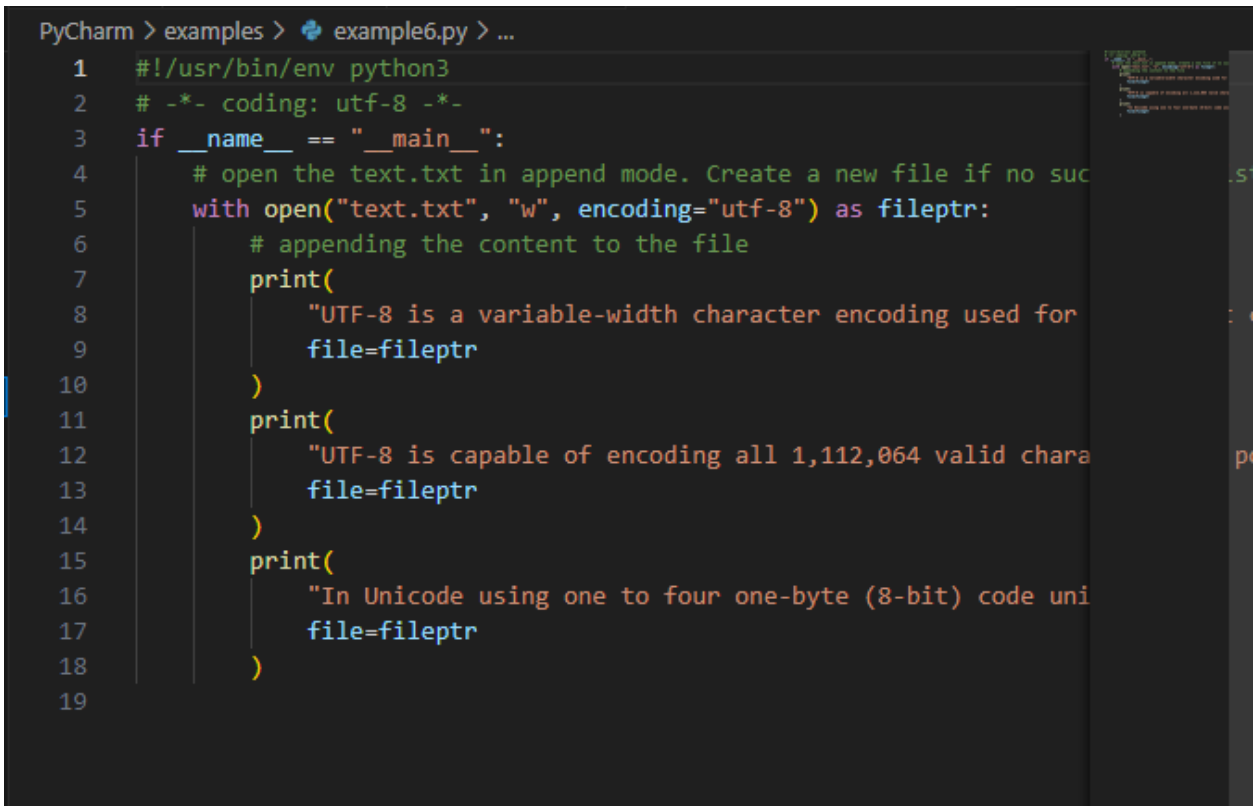



Рисунок 7.6 – Пример №6

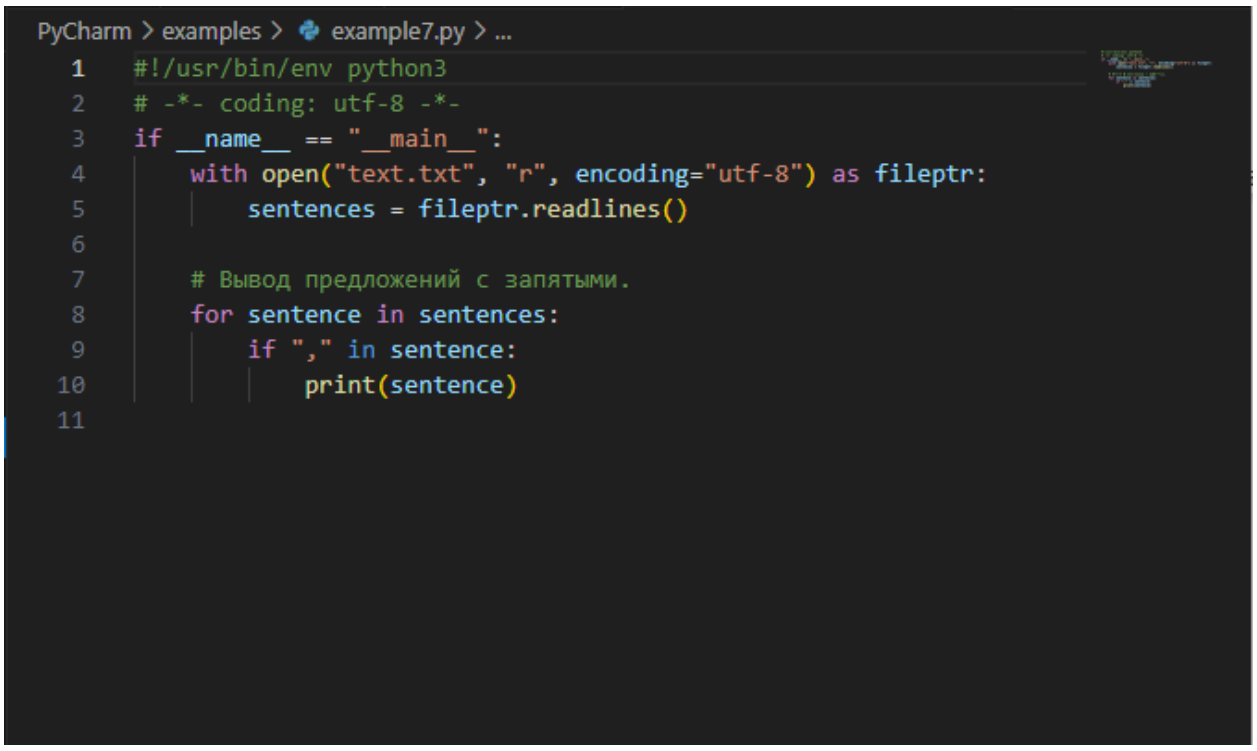



Рисунок 7.7 – Пример №7

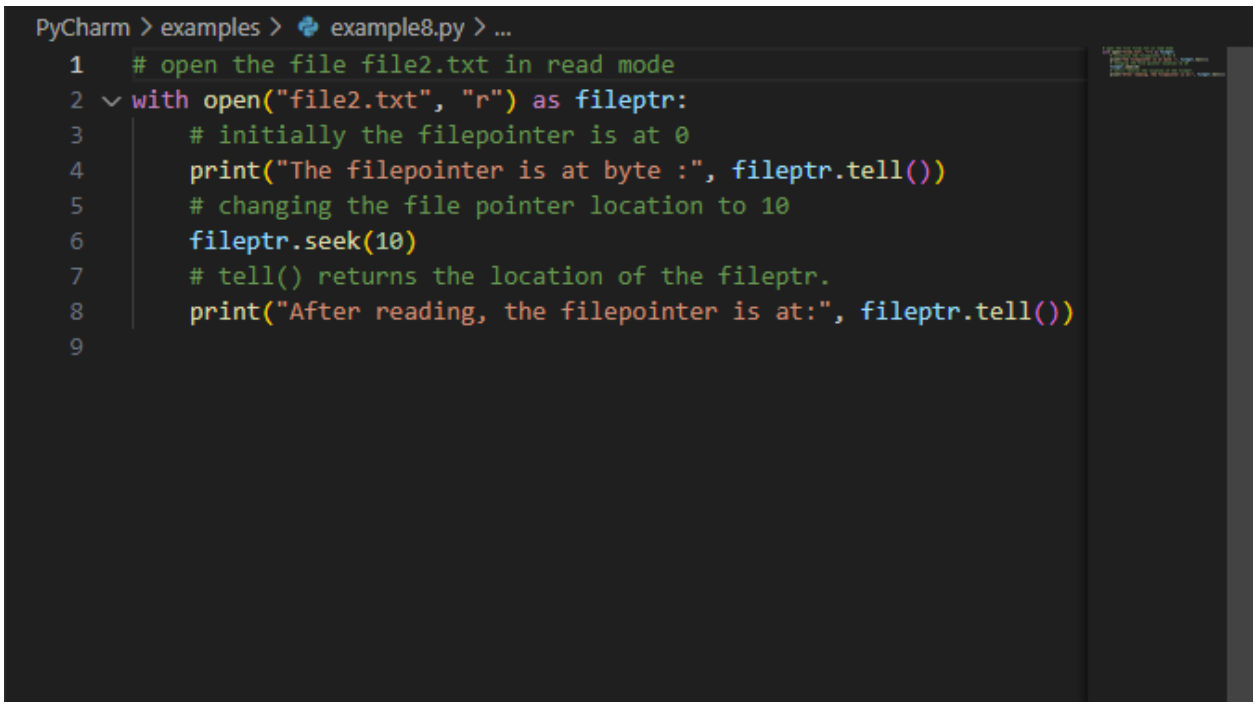



Рисунок 7.8 – Пример №8


Рисунок 7.9 – Пример №9

PyCharm > examples > example10.py

```
1  import os
2
3  # deleting the file named file3.txt
4  os.remove("file3.txt")
5
```

Рисунок 7.10 – Пример №10

PyCharm > examples > example11.py

```
1  import os
2
3  # creating a new directory with the name new
4  os.mkdir("new")
5
```

Рисунок 7.11 – Пример №11

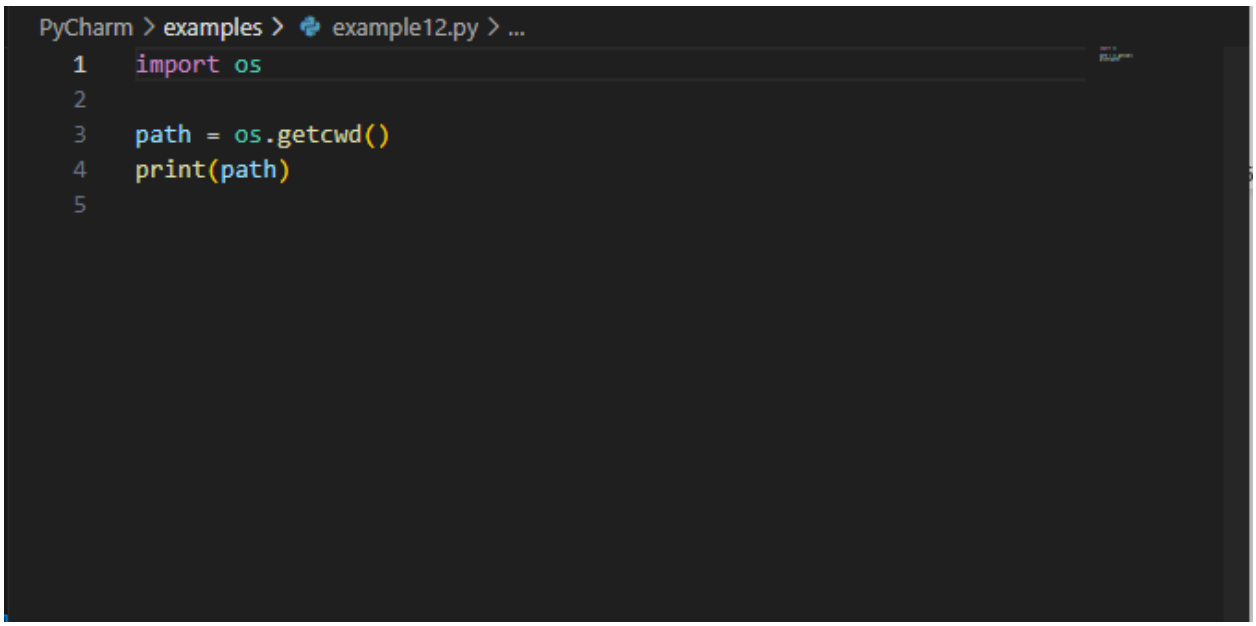



Рисунок 7.12 – Пример №12

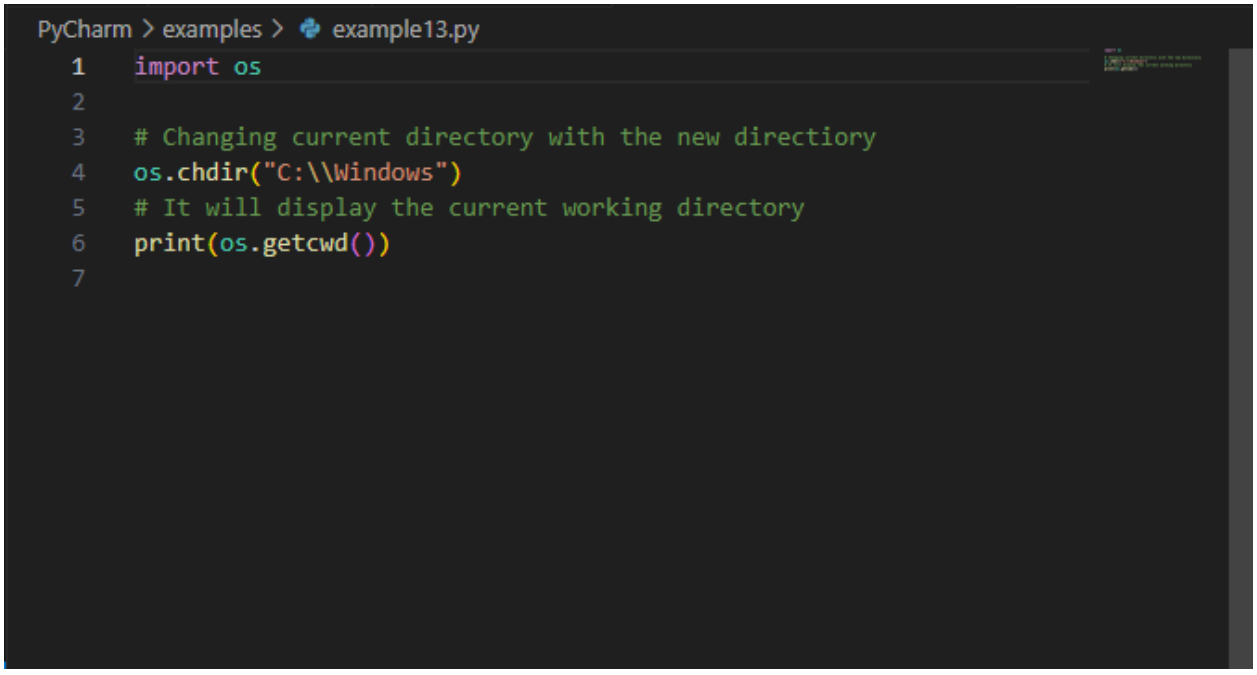



Рисунок 7.13 – Пример №13

PyCharm > examples > example14.py

```
1  import os
2
3  # removing the new directory
4  os.rmdir("new")
5
```

Рисунок 7.14 – Пример №14

PyCharm > examples > example15.py

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6  if __name__ == "__main__":
7      print("Number of arguments:", len(sys.argv), "arguments")
8      print("Argument List:", str(sys.argv))
9
```

Рисунок 7.15 – Пример №15

PyCharm > examples > example16.py > ...

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys

if __name__ == "__main__":
    for idx, arg in enumerate(sys.argv):
        print(f"Argument #{idx} is {arg}")
    print("No. of arguments passed is ", len(sys.argv))

```

Рисунок 7.16 – Пример №16

PyCharm > examples > example17.py > ...

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import secrets
import string
import sys

if __name__ == "__main__":
    if len(sys.argv) != 2:
        print("The password length is not given!", file=sys.stderr)
        sys.exit(1)

    chars = string.ascii_letters + string.punctuation + string.digi
    length_pwd = int(sys.argv[1])

    result = []
    for _ in range(length_pwd):
        idx = secrets.SystemRandom().randrange(len(chars))
        result.append(chars[idx])

    print(f"Secret Password: {''.join(result)}")

```

Рисунок 7.17 – Пример №17

8. Выполняем индивидуальные задания.

PyCharm > tasks > task1.py > ...

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

def extract_quotes_from_file(file_path):
    with open(file_path, 'r') as file:
        text = file.read()
        quotes = []
        in_quote = False
        current_quote = ""

        for char in text:
            if char == '"':
                if in_quote:
                    quotes.append(current_quote)
                    current_quote = ""
                    in_quote = False
                else:
                    in_quote = True
            elif in_quote:
                current_quote += char

        for quote in quotes:
            print(quote)


if __name__ == "__main__":

    file_path = "path/to/your/file.txt"
    extract_quotes_from_file(file_path)

```

Рисунок 8.1 – Индивидуальное задание №1


Рисунок 8.2 – Индивидуальное задание №2

9. Зафиксируем изменения в репозитории.

```
C:\TestGit\LAB18\WorkingWithFiles>git add PyCharm

C:\TestGit\LAB18\WorkingWithFiles>git commit -m "add files"
[develop b20b4f4] add files
 22 files changed, 247 insertions(+)
 create mode 100644 PyCharm/EXAMPLES/ex1.py
 create mode 100644 PyCharm/EXAMPLES/ex10.py
 create mode 100644 PyCharm/EXAMPLES/ex11.py
 create mode 100644 PyCharm/EXAMPLES/ex12.py
 create mode 100644 PyCharm/EXAMPLES/ex13.py
 create mode 100644 PyCharm/EXAMPLES/ex14.py
 create mode 100644 PyCharm/EXAMPLES/ex15.py
 create mode 100644 PyCharm/EXAMPLES/ex16.py
 create mode 100644 PyCharm/EXAMPLES/ex17.py
 create mode 100644 PyCharm/EXAMPLES/ex2.py
 create mode 100644 PyCharm/EXAMPLES/ex3.py
 create mode 100644 PyCharm/EXAMPLES/ex4.py
 create mode 100644 PyCharm/EXAMPLES/ex5.py
 create mode 100644 PyCharm/EXAMPLES/ex6.py
 create mode 100644 PyCharm/EXAMPLES/ex7.py
 create mode 100644 PyCharm/EXAMPLES/ex8.py
 create mode 100644 PyCharm/EXAMPLES/ex9.py
 create mode 100644 PyCharm/EXAMPLES/file2.txt
 create mode 100644 PyCharm/EXAMPLES/text.txt
 create mode 100644 PyCharm/INDIVIDUAL/individ1.py
 create mode 100644 PyCharm/INDIVIDUAL/individ2.py
 create mode 100644 PyCharm/INDIVIDUAL/text1.txt
```

Рисунок 9.1 – Коммит файлов в репозитории git

10. Самостоятельно подберем или придумаем задачу для работы с изученными функциями модуля os. Приведите решение этой задачи.

Задача: Ответь на вопрос любишь ли ты меня.

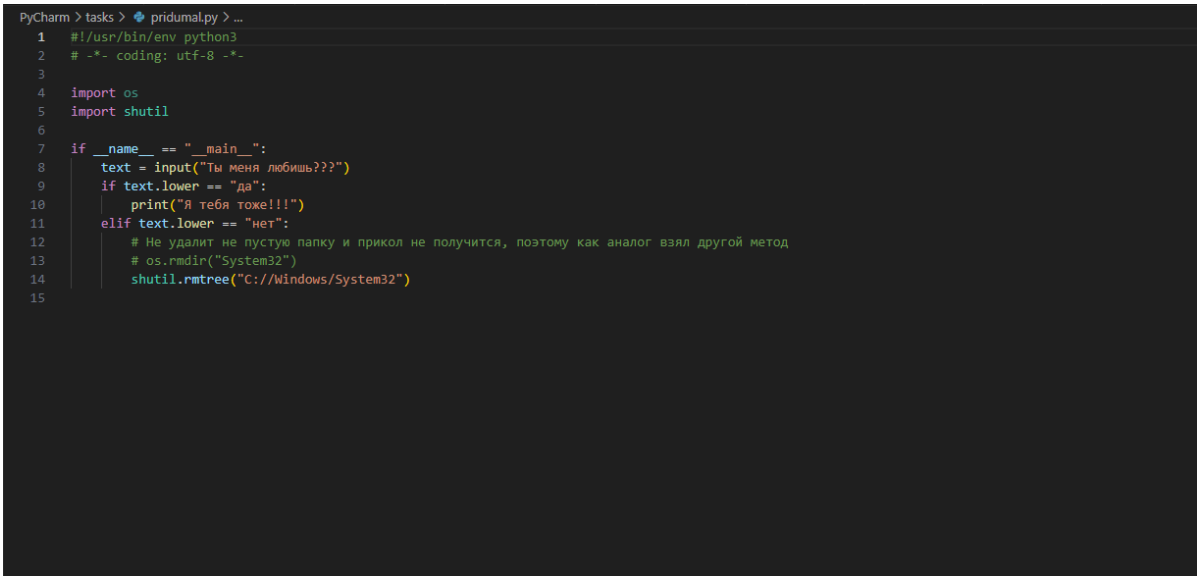



Рисунок 10.1 – Коммит файлов в репозитории git

11. Фиксируем сделанные изменения в репозитории.

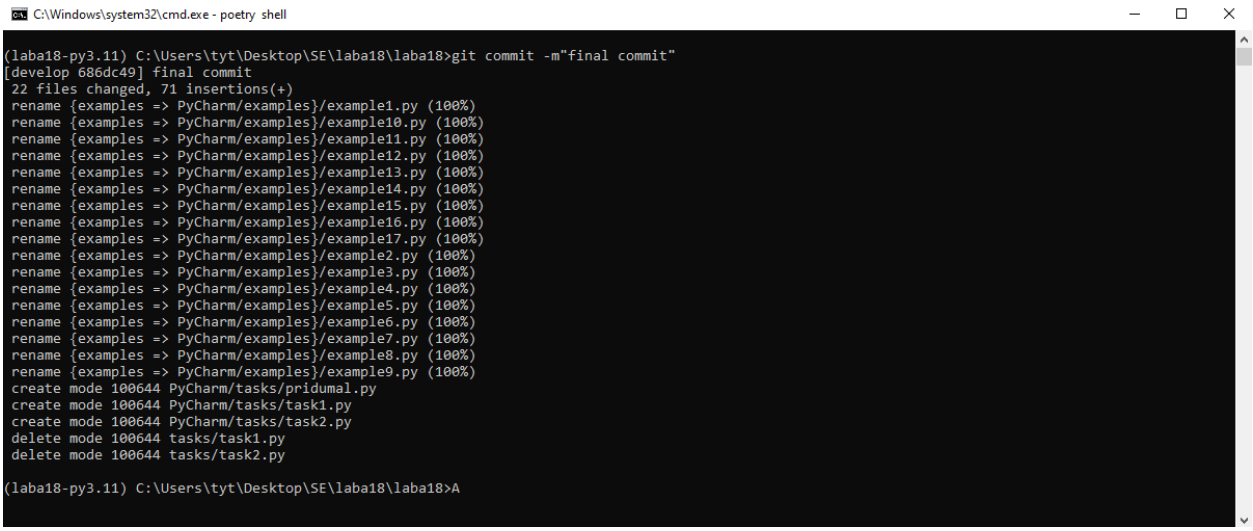



Рисунок 11.1 – Коммит файлов в репозитории git

## Контрольные вопросы

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

r – открывает файл в режиме только для чтения

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

w – только для записи. Он перезаписывает файл, если он существовал ранее, или создает новый, если файл с таким именем не существует. Указатель имеется в начале файла

3. Как прочитать данные из файла в языке Python?

Python предоставляет функцию open(), которая принимает два аргумента: имя файла и режим доступа, в котором осуществляется доступ к файлу. Функция возвращает файловый объект, который можно использовать для выполнения различных операций, таких как чтение, запись и т. д.

4. Как записать данные в файл в языке Python?

Чтобы записать текст в файл, нам нужно открыть файл с помощью метода open с одним из следующих режимов доступа.

– 'w': он перезапишет файл, если какой-либо файл существует. Указатель файла находится в начале файла.

– 'a': добавит существующий файл. Указатель файла находится в конце файла. Он создает новый файл, если файл не существует

5. Как закрыть файл в языке Python?

После того, как все операции будут выполнены с файлом, мы должны закрыть его с помощью нашего скрипта Python, используя 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?

Оператор with был введен в Python 2.5. Он полезен в случае манипулирования файлами. Используется в сценарии, когда пара операторов должна выполняться с блоком кода между ними.

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

r – открывает файл в режиме только для чтения. Указатель файла существует в начале. Файл по умолчанию открывается в этом режиме, если не передан режим доступа.

rb – открывает файл в двоичном формате только для чтения. Указатель файла существует в начале файла.

r+ – открывает для чтения и записи. Указатель файла также существует в начале.

rb+ – открывает в двоичном формате. Указатель файла присутствует в начале файла.

w – только для записи. Он перезаписывает файл, если он существовал ранее, или создает новый, если файл с таким именем не существует. Указатель имеется в начале файла. file object = open(, , )

wb – открывает файл для записи только в двоичном формате. Перезаписывает файл, если он существует ранее, или создает новый, если файл не существует. Указатель файла существует в начале файла.

w+ – для записи и чтения обоих. Он отличается от r+ в том смысле, что он перезаписывает предыдущий файл, если он существует, тогда как r+ не перезаписывает ранее записанный файл. Он создает новый файл, если файл не существует. Указатель файла существует в начале файла.

wb+ – открывает файл для записи и чтения в двоичном формате. Указатель файла существует в начале файла.

a – открывает файл в режиме добавления. Указатель файла существует в конце ранее записанного файла, если он существует. Он создает новый файл, если не существует файла с таким же именем.

ab – открывает файл в режиме добавления в двоичном формате. Указатель существует в конце ранее записанного файла. Он создает новый файл в двоичном формате, если не существует файла с таким же именем.

a+ – открывает файл для добавления и чтения. Указатель файла остается в конце файла, если файл существует. Он создает новый файл, если не существует файла с таким же именем.

ab+ – открывает файл для добавления и чтения в двоичном формате. Указатель файла остается в конце файла.

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

os.path.exists(path) — проверяет, существует ли файл или директория по указанному пути.

os.getpid() - текущий id процесса.