

# Лабораторная работа №7

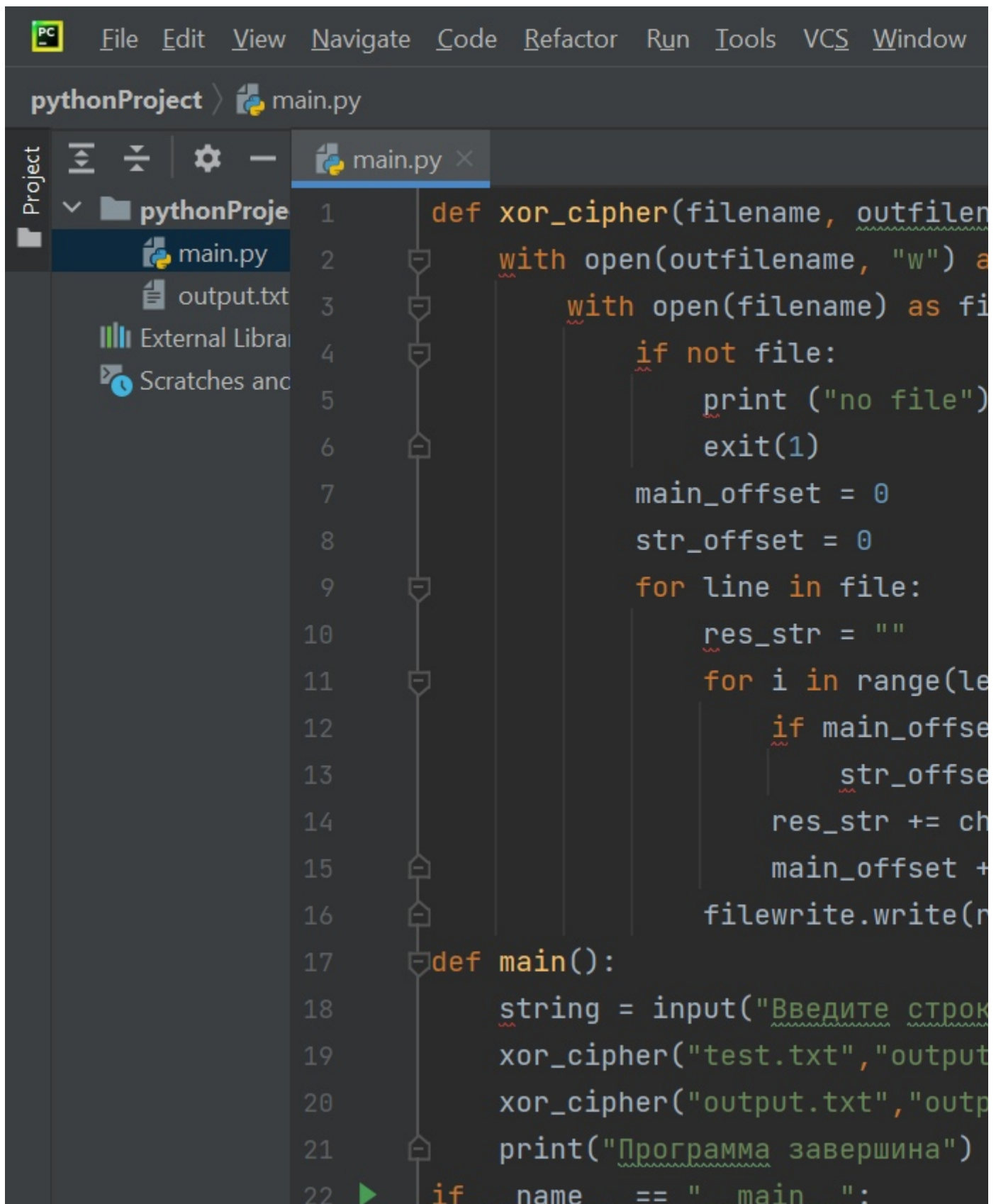
Уваров Илья НПИбд-02-19

## Цель работы

Освоить на практике применение режима однократного гаммирования.

### **Выполнение лабораторной работы**

1. Написал код на языке python, позволяющий шифровать и дешифровать данные в режиме однократного гаммирования.(рис. 1).



The image shows a screenshot of an IDE with a dark theme. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, and Window. The project name 'pythonProject' is visible in the top left. The left sidebar shows a project tree with 'pythonProject' containing 'main.py' and 'output.txt'. The main editor window displays the code for 'main.py'. The code defines a function 'xor\_cipher' that takes 'filename' and 'outfilename' as arguments. It opens 'outfilename' in write mode and 'filename' in read mode. If 'filename' does not exist, it prints 'no file' and exits. It then initializes 'main\_offset' and 'str\_offset' to 0. A loop iterates over each line in 'filename', and an inner loop iterates over each character in the line. For each character, it calculates the XOR with the corresponding character in 'main\_offset' and writes the result to 'outfilename'. The 'main' function prompts the user to enter a string, calls 'xor\_cipher' twice (once with 'test.txt' and 'output.txt', and once with 'output.txt' and 'output.txt'), and prints 'Программа завершена'. The script is guarded by 'if \_\_name\_\_ == "\_\_main\_\_":'.

```
1 def xor_cipher(filename, outfilename):
2     with open(outfilename, "w") as outfile:
3         with open(filename) as file:
4             if not file:
5                 print("no file")
6                 exit(1)
7             main_offset = 0
8             str_offset = 0
9             for line in file:
10                 res_str = ""
11                 for i in range(len(line)):
12                     if main_offset >= len(line):
13                         str_offset = 0
14                         res_str += chr(ord(line[i]) ^ ord('a'))
15                         main_offset += 1
16                 outfile.write(res_str + "\n")
17 def main():
18     string = input("Введите строку: ")
19     xor_cipher("test.txt", "output.txt")
20     xor_cipher("output.txt", "output.txt")
21     print("Программа завершена")
22 if __name__ == "__main__":
```

# Выводы

Освоил на практике применение режима однократного гаммирования.