БЕЛОРУССКИЙ  ГОСУДАРСТВЕННЫЙ  УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Кафедра технологий программирования

**﻿﻿﻿Своя программа для удаления файлов**

Домашнее задание

Шибко Татьяны Александровны

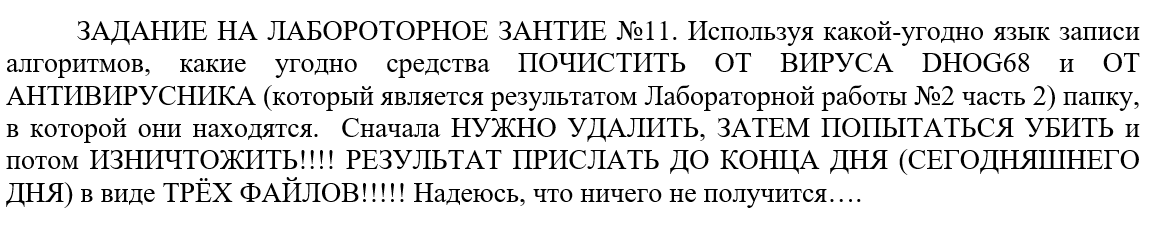
студентки 4 курса 12 группы

специальность "прикладная информатика"

|  |  |
| --- | --- |
| Дата окончания работы 06.10.2024 г. | Преподаватель: |
|  | Доцент К.А. Зубович |

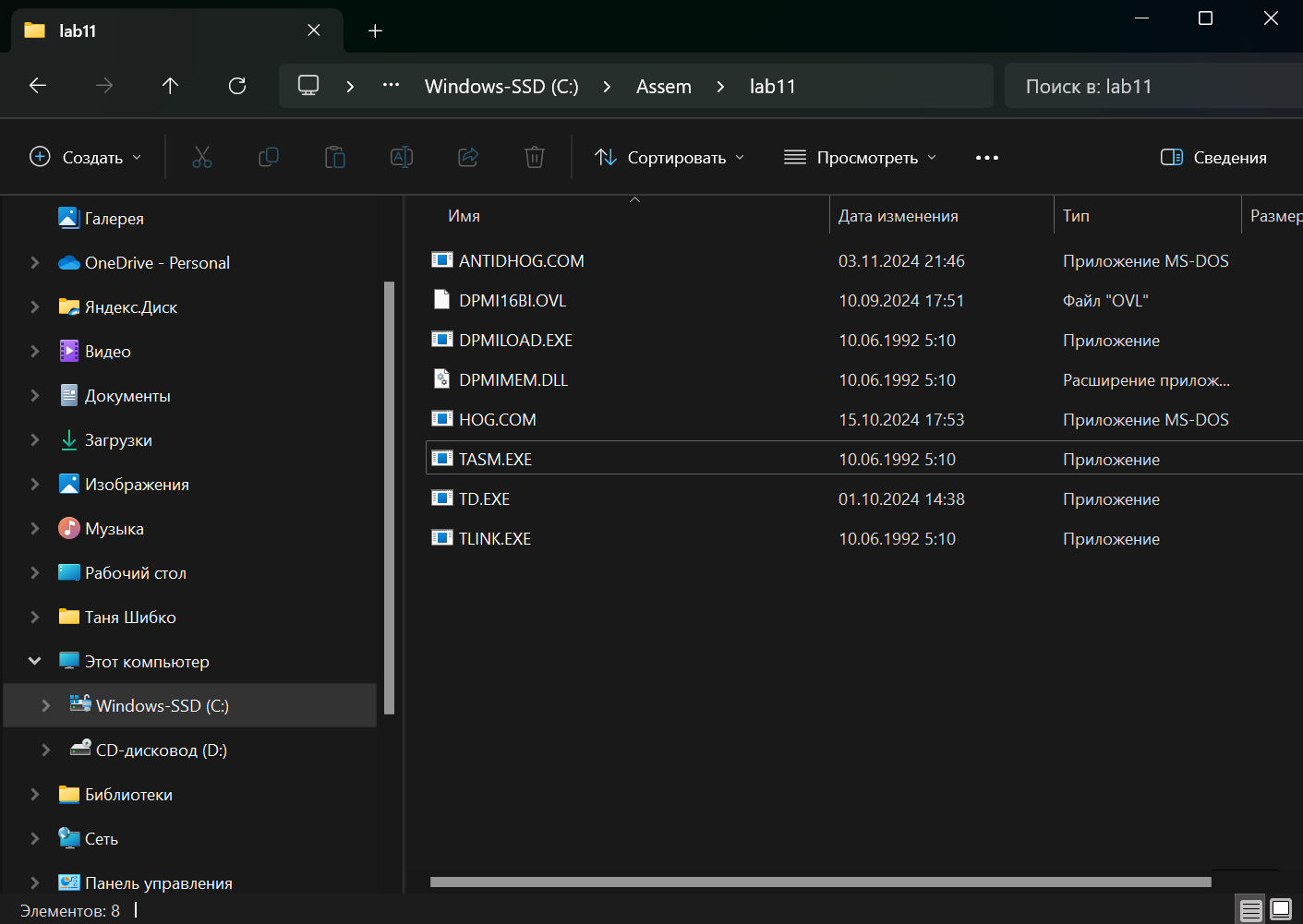
Минск, 2024

**Постановка задания:**



**Ход решения:**

Для начала я создала отдельную папку, где у меня лежат вирус и антивирус (HOG.COM и ANTIDHOG.COM). Именно там я и буду работать дальше.



Теперь перейдём к самому коду:

import os

import signal

import psutil

# Задайте путь к папке

folder\_path = r"C:\Assem\lab11"

# Список зараженных расширений

infected\_extensions = {".obj", ".com", ".exe", ".map", ".asm", ".ovl", ".dll", ".txt", ".lib"}

# Функция для завершения процессов HOG и ANTIDHOG

def kill\_priority\_processes():

    print("Попытка завершения процессов HOG и ANTIDHOG...")

    for proc in psutil.process\_iter(attrs=['pid', 'name']):

        if "HOG" in proc.info['name'] or "ANTIDHOG" in proc.info['name']:

            try:

                os.kill(proc.info['pid'], signal.SIGTERM)

                print(f"Завершен процесс: {proc.info['name']} (PID: {proc.info['pid']})")

            except Exception as e:

                print(f"Не удалось завершить процесс {proc.info['name']}. Ошибка: {e}")

# Функция для приоритетного уничтожения файлов HOG и ANTIDHOG

def destroy\_priority\_files():

    priority\_files = ["HOG.COM", "ANTIDHOG.COM"]

    print("Попытка уничтожения приоритетных файлов HOG и ANTIDHOG...")

    for file\_name in priority\_files:

        file\_path = os.path.join(folder\_path, file\_name)

        if os.path.exists(file\_path):

            try:

                os.chmod(file\_path, 0o777)  # Снятие защиты от записи

                with open(file\_path, 'r+b') as f:

                    f.seek(0)

                    f.write(os.urandom(os.path.getsize(file\_path)))  # Перезапись случайными данными

                os.remove(file\_path)

                print(f"Уничтожен приоритетный файл: {file\_path}")

            except Exception as e:

                print(f"Не удалось уничтожить файл {file\_path}. Ошибка: {e}")

# Функция для уничтожения всех зараженных файлов по расширениям

def destroy\_infected\_files():

    print("Попытка уничтожения всех зараженных файлов...")

    all\_files\_normal = True  # Флаг для отслеживания, были ли зараженные файлы

    for root, dirs, files in os.walk(folder\_path):

        for file\_name in files:

            file\_path = os.path.join(root, file\_name)

            # Проверка расширения файла

            if any(file\_name.endswith(ext) for ext in infected\_extensions):

                try:

                    os.chmod(file\_path, 0o777)

                    with open(file\_path, 'r+b') as f:

                        f.seek(0)

                        f.write(os.urandom(os.path.getsize(file\_path)))

                    os.remove(file\_path)

                    print(f"Уничтожен файл: {file\_path}")

                    all\_files\_normal = False  # Найден зараженный файл

                except Exception as e:

                    print(f"Не удалось уничтожить файл: {file\_path}. Ошибка: {e}")

            else:

                print(f"Файл в норме: {file\_path}")  # Файл не заражен

    if all\_files\_normal:

        print("Все файлы в норме, зараженных файлов не найдено.")

# Основная функция

def main():

    # Завершение процессов HOG и ANTIDHOG

    kill\_priority\_processes()

    # Уничтожение приоритетных файлов HOG и ANTIDHOG

    destroy\_priority\_files()

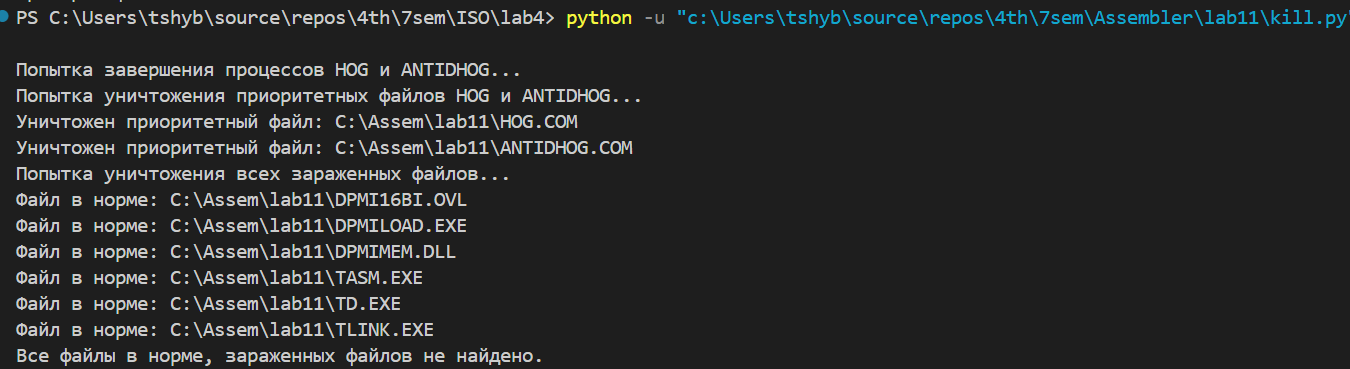
    # Уничтожение всех остальных зараженных файлов

    destroy\_infected\_files()

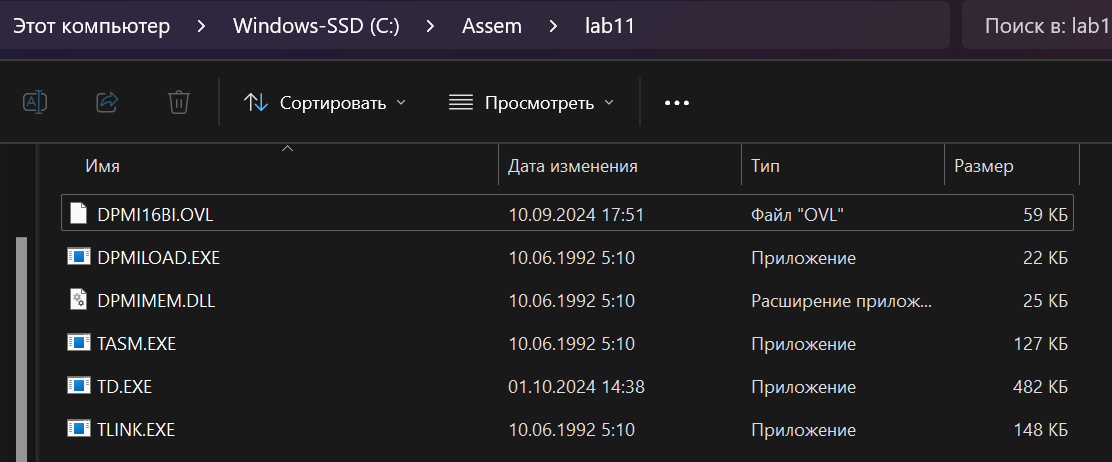
if \_\_name\_\_ == "\_\_main\_\_":

    main()

Запускаем программу и смотрим, что получилось:

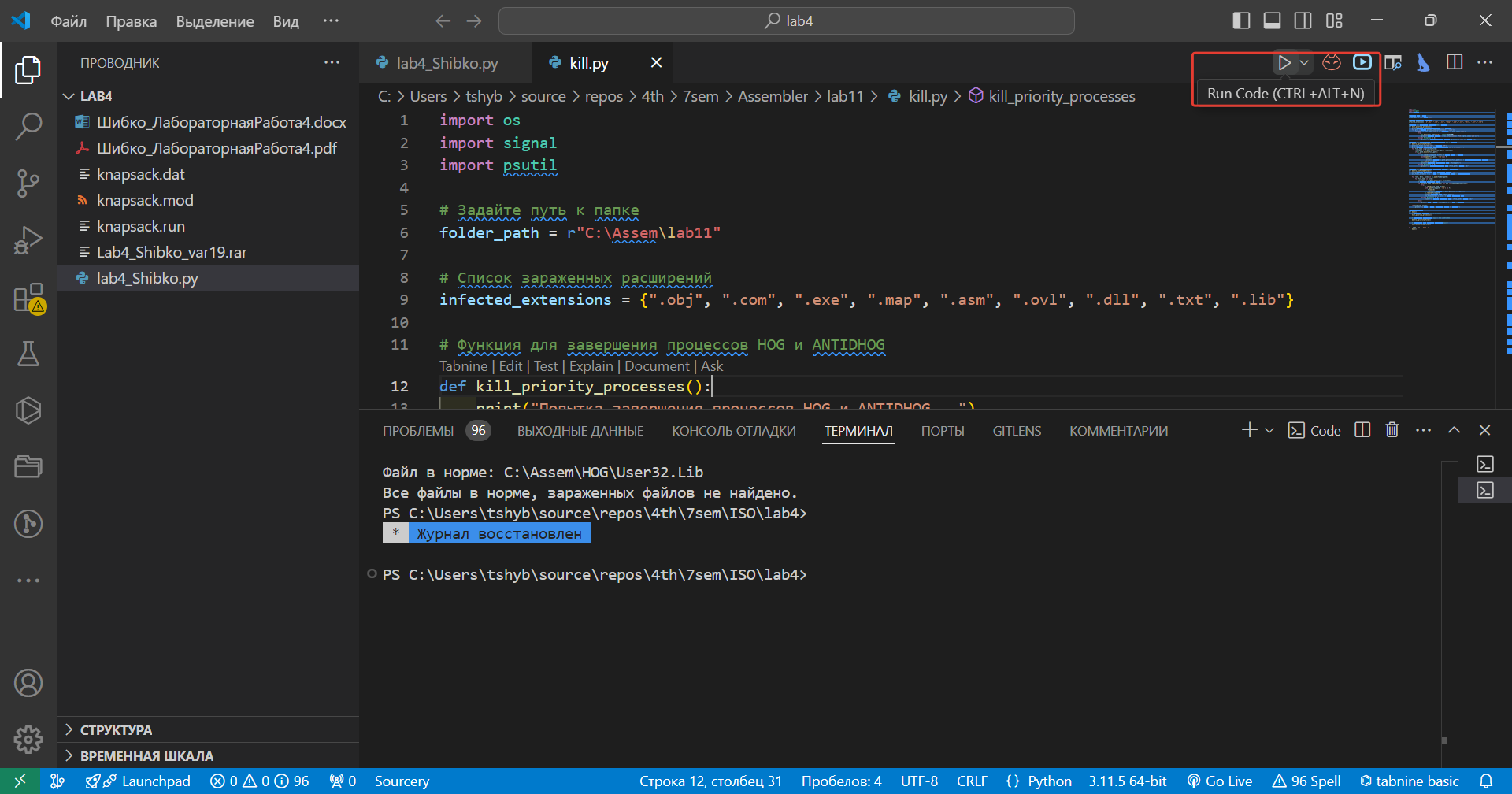


Посмотрим, что у нас в папке lab11



Файл .py является и исходным кодом, и исполнимым модулем одновременно.

Я использовала Visual Studio Code. Вам достаточно просто загрузить файл и открыть его.



Нажимаете сверху на Run Code и в терминале у вас будет результат. Только не забудьте изменить путь к папке:

# Задайте путь к папке

folder\_path = r"C:\Assem\lab11"