

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра экономической информатики

Лабораторная работа №3

«Работа в файлами»

Выполнил:

студент гр. 414302
К. С. Черненко

Проверил:

Мацокин М П.

Минск 2025

Цель работы: закрепление на практике теоретических знаний по работе с файлами на языке программирования Python путем решения комплекса задач.

Вариант 29

Задание 1.

1. Создать программный файл F1 в текстовом формате, записать в него построчно данные, вводимые пользователем. Об окончании ввода данных будет свидетельствовать пустая строка. Скопировать из файла F1 в файл F2 строки, начиная с N до K. Подсчитать количество согласных букв в файле F2.

Листинг 1

```
def create_file_f1():
    with open("F1.txt", "w", encoding="utf-8") as file:
        while (line := input()) != "":
            file.write(line + "\n")

def copy_lines_n_to_k(name_file1, name_file2):
    n, k = int(input("First line: ")), int(input("Second line: "))

    with open(name_file1, "r", encoding="utf-8") as f1:
        lines = f1.readlines()

    if n > k or n < 1 or k > len(lines):
        print("Ошибка в диапазоне строк")
        return

    with open(name_file2, "w", encoding="utf-8") as f2:
        f2.writelines(lines[n - 1:k])

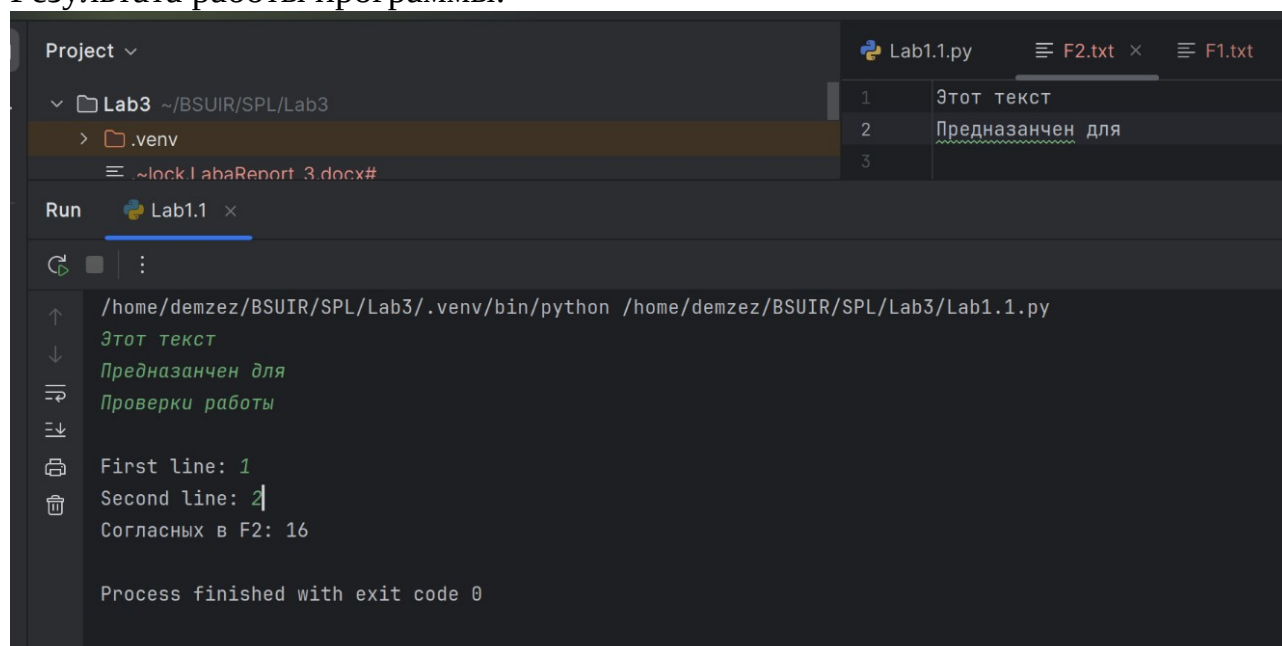
def count_consonants_in_f2():
    consonants = "бвгджзйклмнпрстфхцчшщbcd fghijklmnpqrstvwxyz"

    with open("F2.txt", "r", encoding="utf-8") as file:
        consonant_count = sum(1 for char in file.read().lower() if char in consonants)

    print(f"Согласных в F2: {consonant_count}")

create_file_f1()
copy_lines_n_to_k("F1.txt", "F2.txt")
count_consonants_in_f2()
```

Результата работы программы:



```
Project ▾
  ▾ Lab3 ~\BSUIR\SPL\Lab3
    > .venv
      ~lock\abaReport_3.docx#
Run Lab1.1 x
  ↻
  /home/demzez/BSUIR/SPL/Lab3/.venv/bin/python /home/demzez/BSUIR/SPL/Lab3/Lab1.1.py
  Этот текст
  Предназанчен для
  Проверки работы
  First line: 1
  Second line: 2
  Согласных в F2: 16
  Process finished with exit code 0
```

Задание 2.

Имеется текстовый файл «Клиент банка», строка которого содержит в себе информацию: фамилия клиента, сумма на счете, дата последнего изменения.

Вывести на экран все фамилии, сумма на счету которых больше 1000. Определить клиентов, с максимальной суммой на счету. Файл заполнить заранее (не программно).

Пример файла:

Иванов 120 12.09.2022

Петров 0 15.08.2022

Листинг 2

```
def process_bank_clients():
    best_clients = []
    with open("bank_clients.txt", "r") as bank_clients_file:
        best_client = bank_clients_file.readline().split()

    for line in bank_clients_file:
        data_line = line.split()

        if int(best_client[1]) < int(data_line[1]):
            best_client = data_line

        if int(data_line[1]) > 1000:
            best_clients.append(' '.join(data_line))

    print("Лучший клиент - " + ' '.join(best_client))
```

```
print("Список всех богатых:\n" + "\n".join(best_clients))

process_bank_clients()
```

Результата работы программы:

The screenshot shows an IDE with two panes. The top pane displays the output of a program, listing seven clients with their names, wealth, and dates. The bottom pane shows the command prompt output, which identifies the richest client and lists the top three richest clients.

№	Имя	Сумма	Дата
1	Иванов	120	12.09.2022
2	Петров	0	15.08.2022
3	Сидоров	1500	20.09.2022
4	Кузнецов	800	05.09.2022
5	Смирнов	2500	18.09.2022
6	Попов	300	10.09.2022
7	Васильев	1800	22.09.2022

Output in the terminal:

```
/home/demzez/BSUIR/SPL/Lab3/.venv/bin/python /home/demzez/BSUIR/SPL/Lab3/Lab1.2.py
Лучший клиент - Смирнов 2500 18.09.2022
Список всех богатых:
Сидоров 1500 20.09.2022
Смирнов 2500 18.09.2022
Васильев 1800 22.09.2022
Process finished with exit code 0
```

Задание 3.

Сформировать (не программно) текстовый файл. В нём каждая строка должна описывать учебный предмет и наличие лекционных, практических и лабораторных занятий по предмету. Сюда должно входить и количество занятий. Необязательно, чтобы для каждого предмета были все типы занятий.

Сформировать словарь, содержащий название предмета и общее количество занятий по нему. Вывести его на экран.

Примеры строк файла:

Информатика: 100(л) 50(пр) 20(лаб).

Физика: 30(л) 10(лаб)

Физкультура: 30(пр)

Пример словаря: {"Информатика": 170, "Физика": 40, "Физкультура": 30}

Листинг 3

```
def count_subject_hours():
    subjects = {}

    with open("subjects.txt", "r", encoding="utf-8") as file:
```

```

for line in file:

    parts = line.split(':')
    subject_name = parts[0].strip()
    hours_str = parts[1]

    total_hours = 0
    for part in hours_str.split():
        if '(' in part:
            hours = part.split('(')[0]
            total_hours += int(hours)

    subjects[subject_name] = total_hours

print(subjects)

count_subject_hours()

```

Результата работы программы:

```

Project
└─ Lab3 ~/BSUIR/SPL/Lab3
   └─ .venv
      ├── .lock.LabaReport_3.docx#
      ├── bank_clients.txt
      └── firms.txt

Run
Lab1.3 x
/home/demzez/BSUIR/SPL/Lab3/.venv/bin/python /home/demzez/BSUIR/SPL/Lab3/Lab1.3.py
{'Информатика': 170, 'Физика': 40, 'Физкультура': 30, 'Математика': 120, 'Химия': 50}
Process finished with exit code 0

```

Задание 4.

Создать вручную и заполнить несколькими строками текстовый файл, в котором каждая строка будет содержать данные о фирме: название, форма собственности, выручка, издержки.

Пример строки файла: firm_1 ООО 10000 5000.

Необходимо построчно прочитать файл, вычислить прибыль каждой компании, а также среднюю прибыль. Если фирма получила убытки, в расчёт средней прибыли её не включать.

Далее реализовать список. Он должен содержать словарь с фирмами и их прибылями, а также словарь со средней прибылью. Если фирма получила убытки, также добавить её в словарь (со значением убытков).

Пример списка: [{"firm_1": 5000, "firm_2": 3000, "firm_3": 1000}, {"average_profit": 2000}].

Итоговый список сохранить в виде json-объекта в соответствующий файл.

Пример json-объекта:

[{"firm_1": 5000, "firm_2": 3000, "firm_3": 1000}, {"average_profit": 2000}]

Листинг 4

```
import json

def calculate_firm_profit():
    firms_profit = {}
    total_profit = 0
    profitable_firms = 0

    with open("firms.txt", "r", encoding="utf-8") as file:
        for line in file:
            data = line.split()
            name = data[0]
            revenue = int(data[2])
            costs = int(data[3])

            profit = revenue - costs
            firms_profit[name] = profit

            # Считаем среднюю прибыль только для прибыльных фирм
            if profit > 0:
                total_profit += profit
                profitable_firms += 1

    average_profit = total_profit / profitable_firms if profitable_firms > 0 else 0

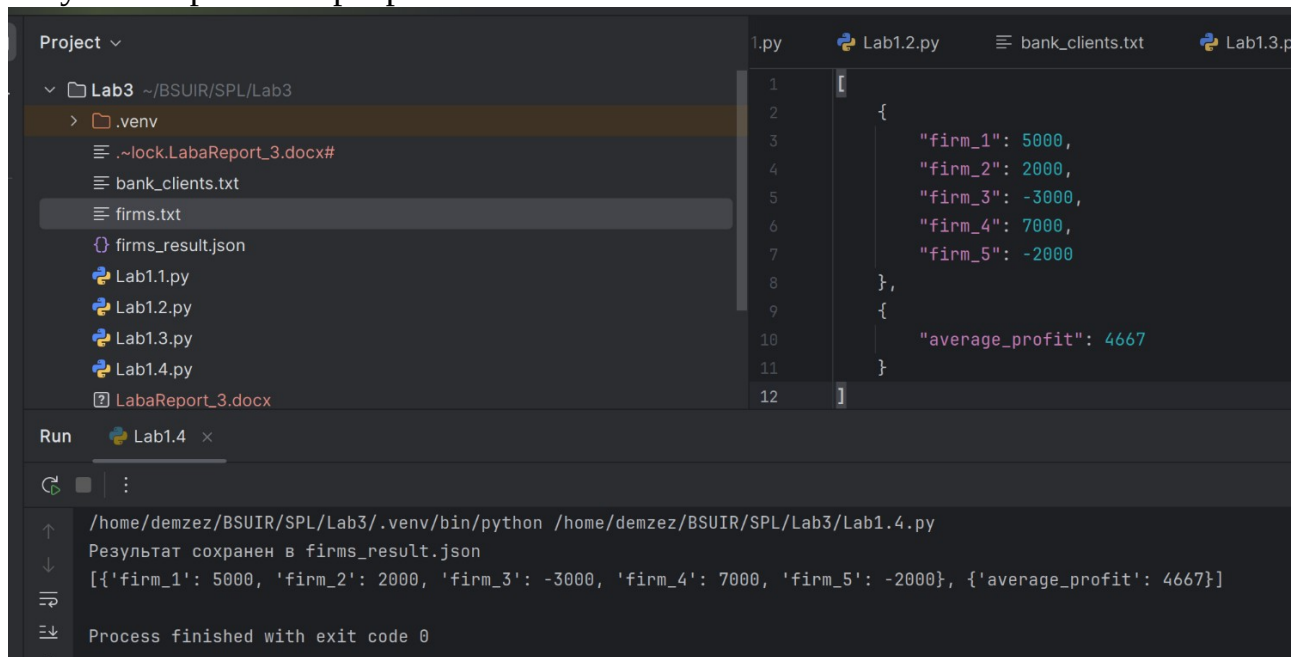
    result_list = [firms_profit, {"average_profit": round(average_profit)}] # округление

    # Сохраняем в JSON файл
    with open("firms_result.json", "w", encoding="utf-8") as json_file:
        json.dump(result_list, json_file, ensure_ascii=False, indent=4)

    print("Результат сохранен в firms_result.json")
    print(result_list)

calculate_firm_profit()
```

Результата работы программы:



The screenshot shows an IDE with a project named 'Lab3'. The file explorer on the left lists files including 'bank_clients.txt', 'firms.txt', 'firms_result.json', and several 'Lab' files. The main editor displays a Python script (Lab1.4.py) with the following code:

```
1 [
2     {
3         "firm_1": 5000,
4         "firm_2": 2000,
5         "firm_3": -3000,
6         "firm_4": 7000,
7         "firm_5": -2000
8     },
9     {
10        "average_profit": 4667
11    }
12 ]
```

The 'Run' panel at the bottom shows the command executed: `/home/demzez/BSUIR/SPL/Lab3/.venv/bin/python /home/demzez/BSUIR/SPL/Lab3/Lab1.4.py`. The output indicates that the result was saved in 'firms_result.json' and displays the resulting JSON data: `[{'firm_1': 5000, 'firm_2': 2000, 'firm_3': -3000, 'firm_4': 7000, 'firm_5': -2000}, {'average_profit': 4667}]`. The process finished with exit code 0.

Вывод:

В результате выполнения лабораторной работы были успешно закреплено умение работы с файлами на языке программирования Python, освоена основная область применения файлов и их синтаксис работы с ними.