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**

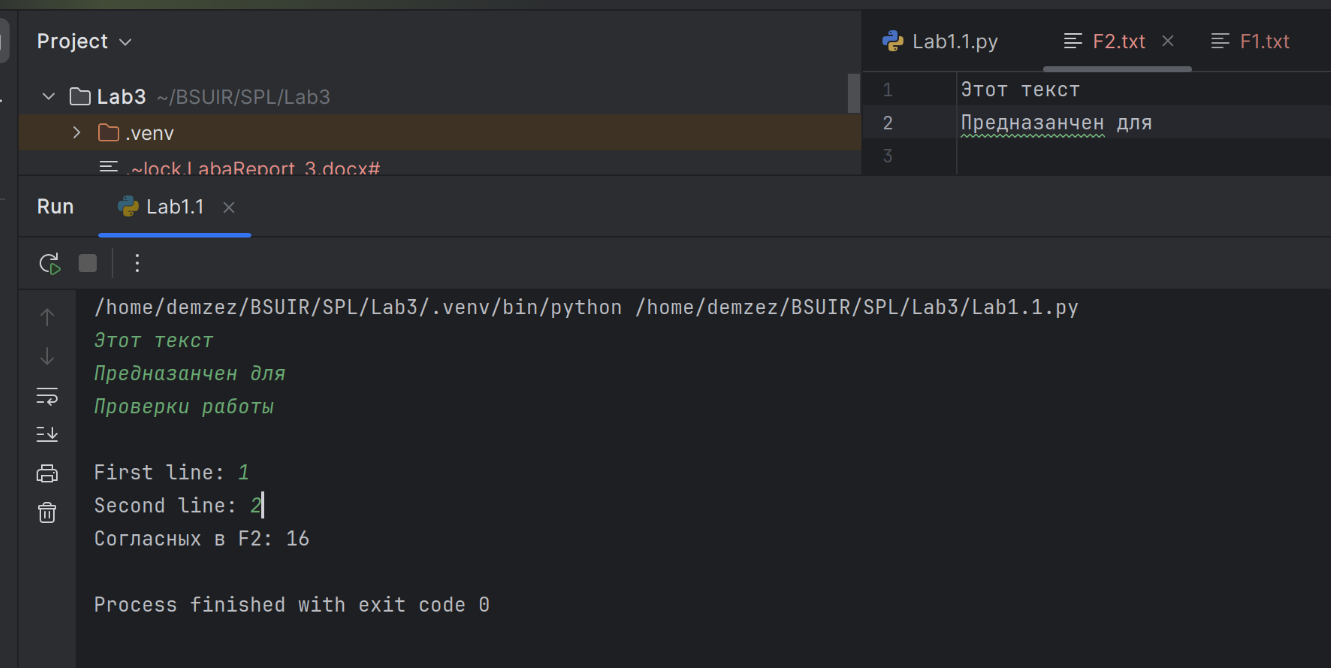
**Задание 1.**

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 = "бвгджзйклмнпрстфхцчшщbcdfghjklmnpqrstvwxyz"   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:



**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:

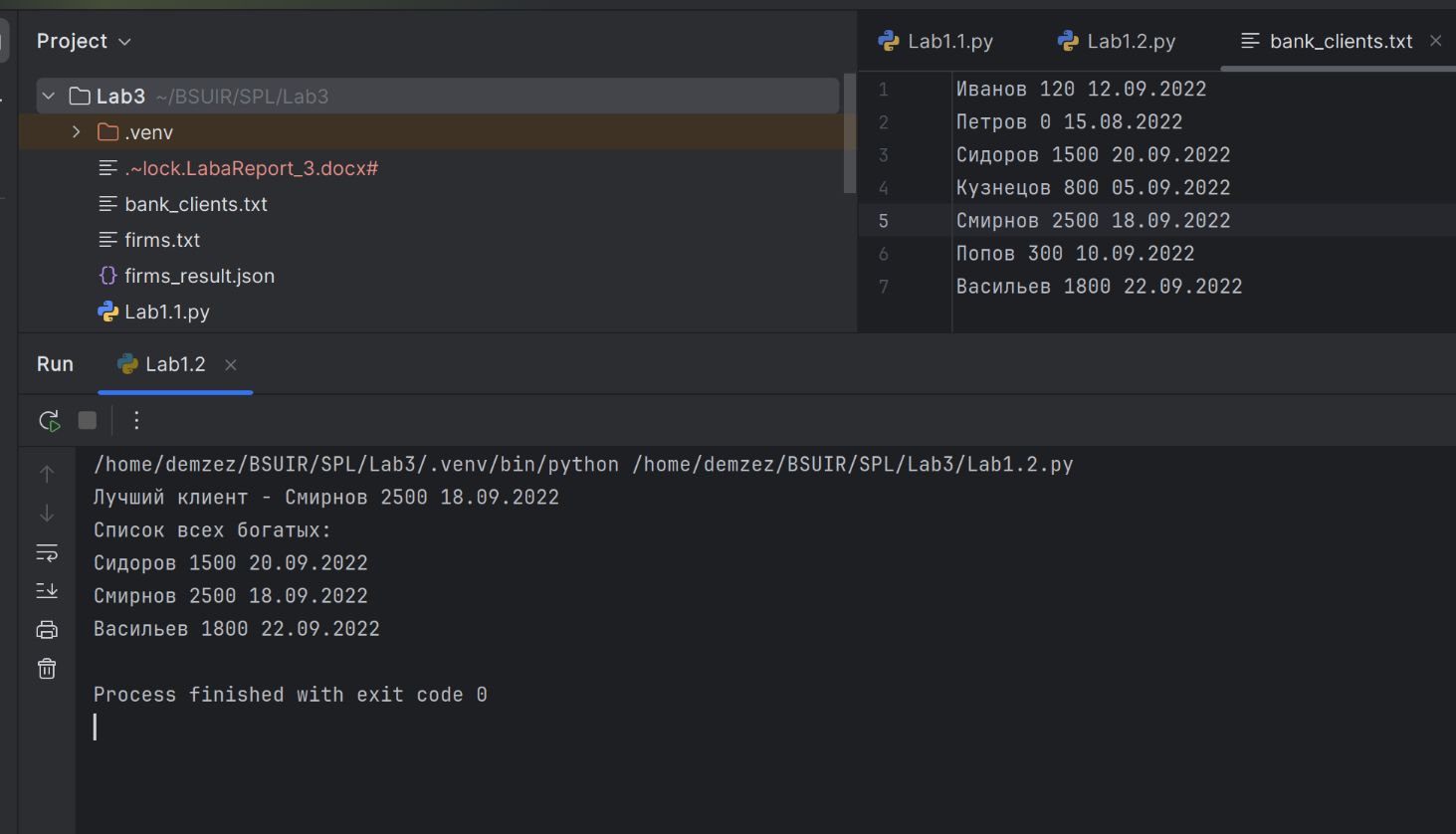
Иванов 120 12.09.2022

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:

****

**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)

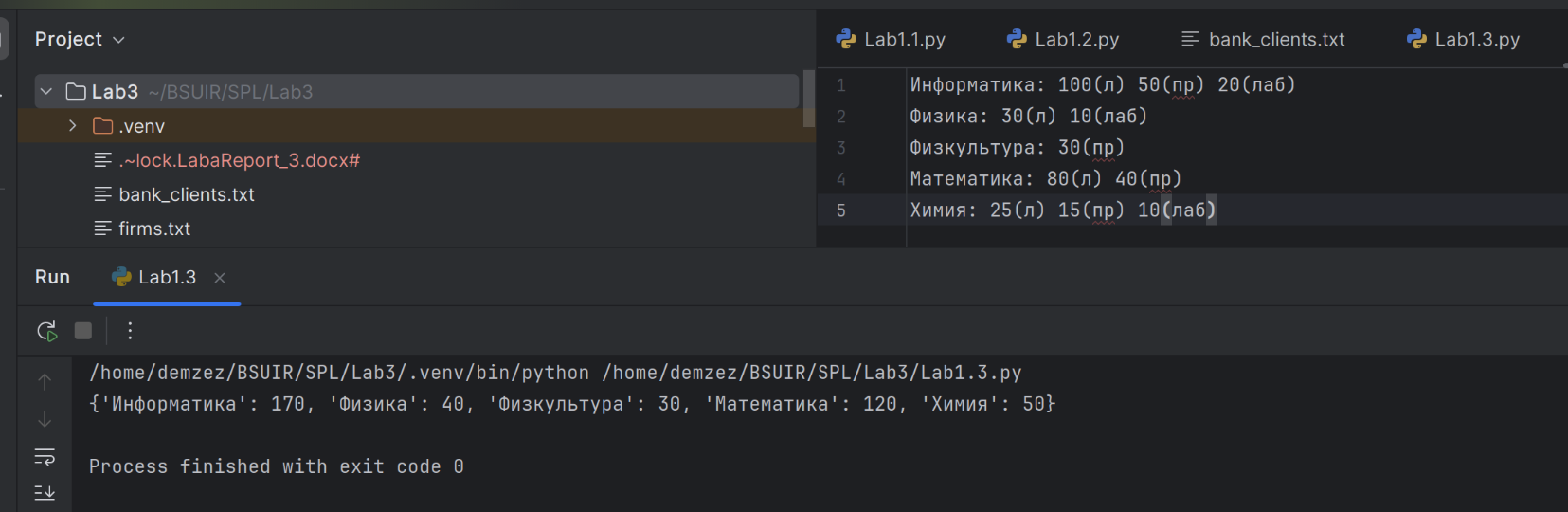
Пример словаря: {“Информатика”: 170, “Физика”: 40, “Физкультура”:

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:



**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:

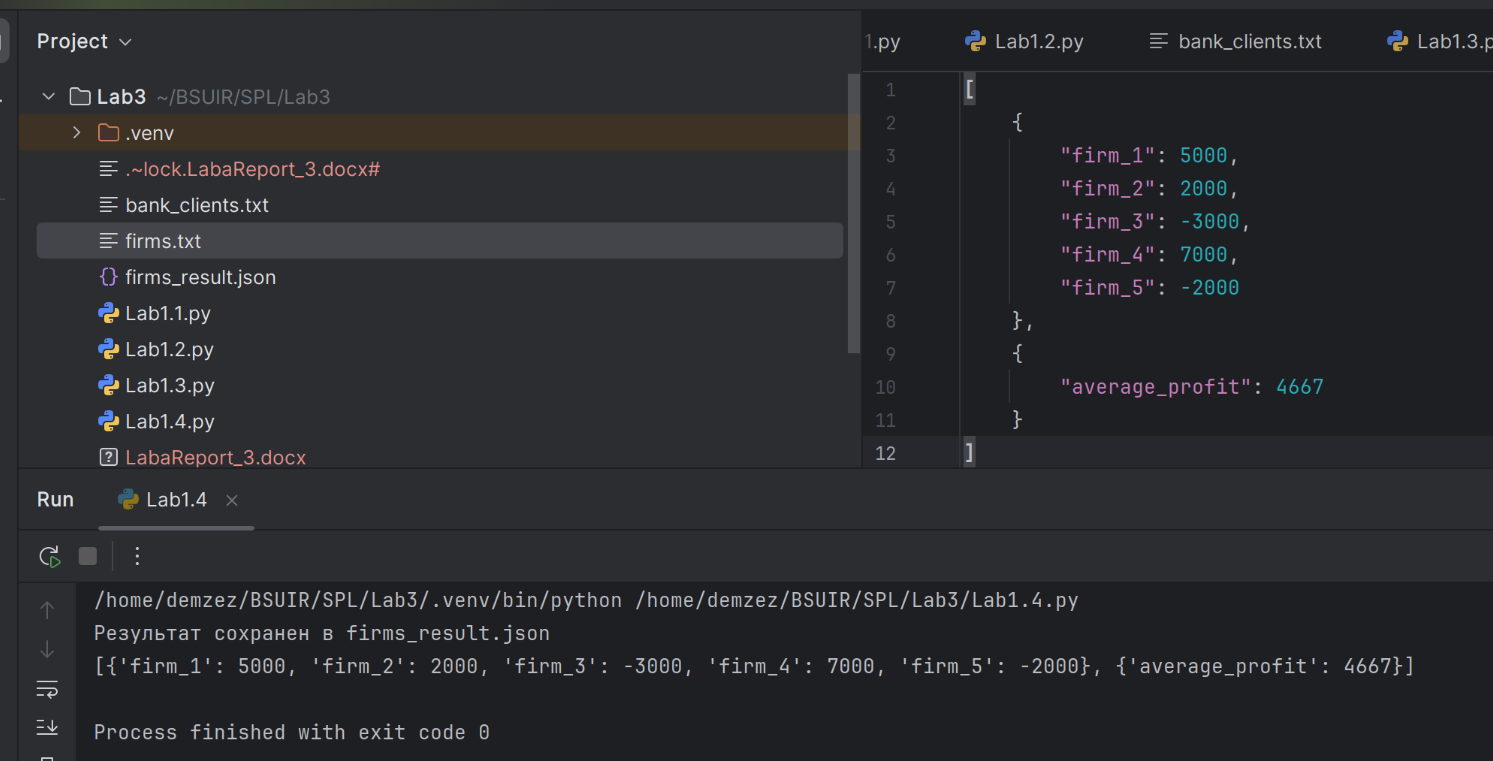
[{"firm\_1": 5000, "firm\_2": 3000, "firm\_3": 1000}, {"average\_profit":

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:



**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.