

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

Отчет по лабораторной работе №0
по курсу «Алгоритмы и структуры данных»
Тема: Работа с файлами

Выполнил:
Марченко В.А.
К3141

Проверил:
Афанасьев А.В.

Санкт-Петербург
2024 г.

Содержание отчета

Содержание отчета	2
Задания по варианту	3
Задание 1. Ввод-вывод	3
Задача $a + b$	3
Задача $a + b^2$	4
Задача $a + b$ с использованием файлов	5
Задача $a + b^2$ с использованием файлов	6
Микро-вывод	7
Задание 2. Число Фибоначчи	8
Микро-вывод	9
Задание 3. Еще про числа Фибоначчи	10
Микро-вывод	11
Задание 4. Тестирование ваших алгоритмов	12
Микро-вывод	13
Вывод	14

Задания по варианту

Задание 1. Ввод-вывод

Задача $a + b$

В данной задаче требуется вычислить сумму двух заданных чисел. Вход: одна строка, которая содержит два целых числа a и b . Для этих чисел выполняются условия $-10^9 \leq a, b \leq 10^9$. Выход: единственное целое число – результат сложения $a + b$.

Листинг кода:

```
a, b = map(int, input().split())
print(a + b)
```

Текстовое объяснение решения:

Сначала программа принимает в себя строку с двумя числами через пробел, после чего разделяет ее и преобразует в числовые значения. Далее происходит вывод результата $a + b$.

Результат работы кода:

```
● vадим@Vадимs-MacBook-Pro asd_labs % /usr/local/bin/python3 "
  /Users/vадим/Desktop/projects/asd_labs/lab0/Task 1/1/1.py"
  10 12
  22
○ vадим@Vадимs-MacBook-Pro asd_labs % □
```

Задача $a + b^2$

В данной задаче требуется вычислить значение $a + b^2$. Вход: одна строка, которая содержит два целых числа a и b . Для этих чисел выполняются условия $-10^9 \leq a, b \leq 10^9$. Выход: единственное целое число – результат сложения $a + b^2$.

Листинг кода:

```
a, b = map(int, input().split())
print(a + b ** 2)
```

Текстовое объяснение решения:

Сначала программа принимает в себя строку с двумя числами через пробел, после чего разделяет ее и преобразует в числовые значения. Далее происходит вывод результата $a + b^2$.

Результат работы кода:

```
● vladim@Vladim-MacBook-Pro asd_labs % /usr/local/bin/python3 "
  /Users/vladim/Desktop/projects/asd_labs/lab0/Task 1/1/1.py"
  2 5
  27
○ vladim@Vladim-MacBook-Pro asd_labs %
```

Задача $a + b$ с использованием файлов

- Имя входного файла: input.txt
- Имя выходного файла: output.txt
- Формат входного файла. Входной файл состоит из одной строки, которая содержит два целых числа a и b . Для этих чисел выполняются условия $-10^9 \leq a, b \leq 10^9$.
- Формат выходного файла. Выходной файл единственное целое число — результат сложения $a + b$.

Листинг кода:

```
import os

f = open(os.path.join(os.path.dirname(__file__), "input.txt"), "r")
a, b = map(int, f.readline().split())

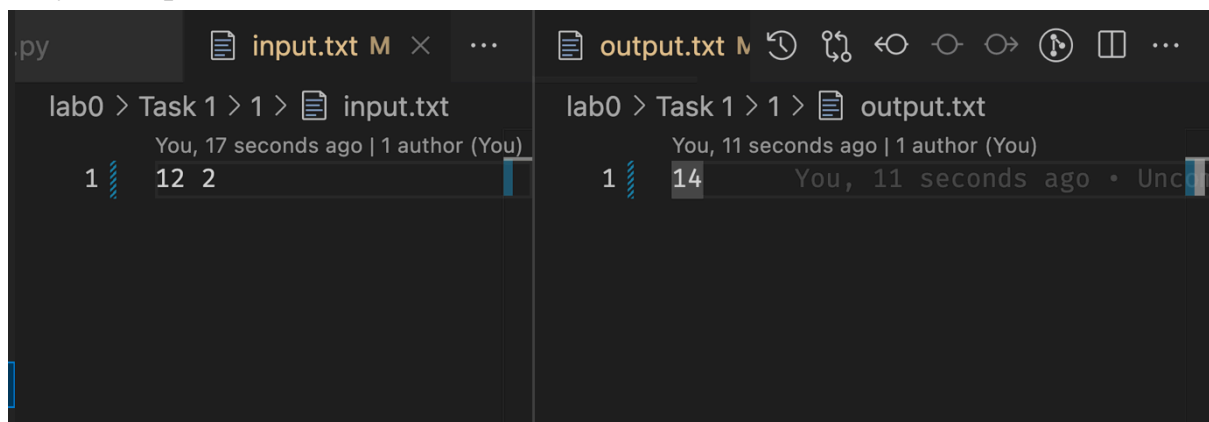
o = open(os.path.join(os.path.dirname(__file__), "output.txt"), "w")
o.write(str(a + b))

f.close()
o.close()
```

Текстовое объяснение решения:

Сначала открываем входной файл input.txt с правами чтения. Чтобы определить его местоположение относительно файла скрипта, используется библиотека os и переменная __file__. После открытия считывается строка, в которой находятся числа, делится на месте пробела — данные преобразовываются в числовые значения. Далее открываем выходной файл output.txt с правами записи и записываем туда значение $a + b$. В конце закрываем файлы, с которыми работали.

Результат работы кода:



Задача $a + b^2$ с использованием файлов

- Имя входного файла: input.txt
- Имя выходного файла: output.txt
- Формат входного файла. Входной файл состоит из одной строки, которая содержит два целых числа a и b . Для этих чисел выполняются условия $-10^9 \leq a, b \leq 10^9$.
- Формат выходного файла. Выходной файл единственное целое число — результат сложения $a + b^2$.

Листинг кода:

```
import os

f = open(os.path.join(os.path.dirname(__file__), "input.txt"), "r")
a, b = map(int, f.readline().split())

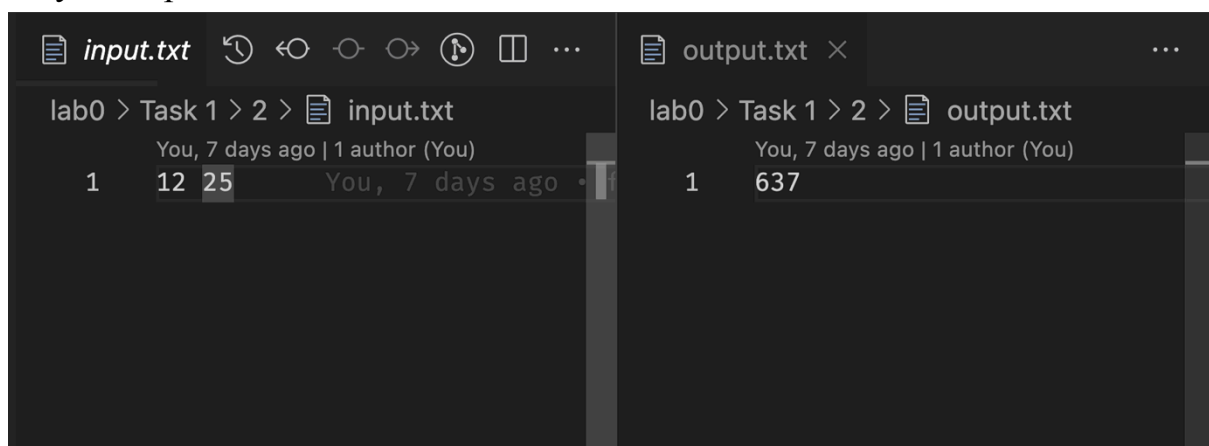
o = open(os.path.join(os.path.dirname(__file__), "output.txt"), "w")
o.write(str(a + b ** 2))

f.close()
o.close()
```

Текстовое объяснение решения:

Сначала открываем входной файл input.txt с правами чтения. Чтобы определить его местоположение относительно файла скрипта, используется библиотека os и переменная __file__. После открытия считывается строка, в которой находятся числа, делится на месте пробела — данные преобразовываются в числовые значения. Далее открываем выходной файл output.txt с правами записи и записываем туда значение $a + b^2$. В конце закрываем файлы, с которыми работали.

Результат работы кода:



Микро-вывод

Таким образом, в этом задании мы вспомнили, как работать с математическими операциями в Python и как работать с файлами.

Задание 2. Число Фибоначчи

Задание – разработать эффективный алгоритм для подсчета чисел Фибоначчи.

- Имя входного файла: input.txt
- Имя выходного файла: output.txt
- Формат входного файла. Целое число n . $0 \leq n \leq 45$.
- Формат выходного файла. Число F_n .

Листинг кода:

```
import os

f = open(os.path.join(os.path.dirname(__file__), "input.txt"), "r")
n = int(f.readline())

o = open(os.path.join(os.path.dirname(__file__), "output.txt"), "w")

if n <= 1:
    o.write(str(n))
else:
    a, b = 0, 1
    for _ in range(2, n + 1):
        a, b = b, a + b

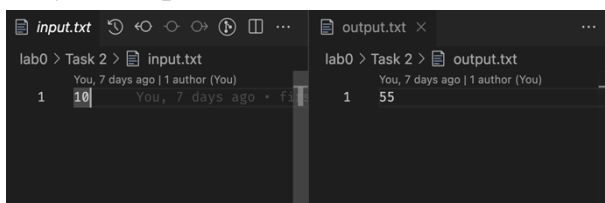
    o.write(str(b))

f.close()
o.close()
```

Текстовое объяснение решения:

Сначала открываем входной файл input.txt с правами чтения. Чтобы определить его местоположение относительно файла скрипта, используется библиотека os и переменная __file__. После открытия считываем строку, в которой находится число и преобразуем данные в числовое значение. Чтобы записать ответ, открываем выходной файл output.txt с правами записи. Если входное число n меньше или равно 1, то сразу записываем ответ в файл. Иначе ищем число Фибоначчи итеративным способом (перезаписываем переменные) и записываем результат в файл. В конце закрываем рабочие файлы.

Результат работы кода:



Микро-вывод

Благодаря это заданию, мы вспомнили последовательность Фибоначчи и как эффективно вычислять число из этой последовательности.

Задание 3. Еще про числа Фибоначчи

Задача – разработать эффективный алгоритм для подсчета чисел Фибоначчи и вывести его последнюю цифру.

- Имя входного файла: input.txt
- Имя выходного файла: output.txt
- Формат входного файла. Целое число n . $0 \leq n \leq 10^7$.
- Формат выходного файла. Одна последняя цифра числа F_n .

Листинг кода:

```
import os

f = open(os.path.join(os.path.dirname(__file__), "input.txt"), "r")
n = int(f.readline())

o = open(os.path.join(os.path.dirname(__file__), "output.txt"), "w")

if n <= 1:
    o.write(str(n))
else:
    a, b = 0, 1
    for _ in range(2, n + 1):
        a, b = b, a + b

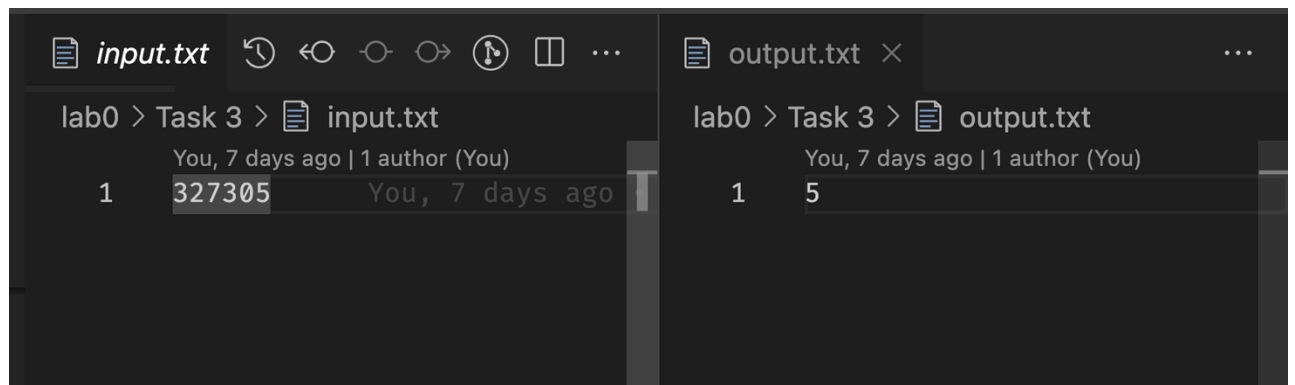
    o.write(str(b % 10))

f.close()
o.close()
```

Текстовое объяснение решения:

Сначала открываем входной файл input.txt с правами чтения. Чтобы определить его местоположение относительно файла скрипта, используется библиотека os и переменная __file__. После открытия считываем строку, в которой находится число и преобразуем данные в числовое значение. Чтобы записать ответ, открываем выходной файл output.txt с правами записи. Если входное число n меньше или равно 1, то сразу записываем ответ в файл. Иначе ищем число Фибоначчи итеративным способом (перезаписываем переменные) и записываем последнюю цифру результата в файл (последнюю цифру можно получить, узнав остаток от деления числа на 10 – $b \% 10$). В конце закрываем рабочие файлы.

Результат работы кода:



The screenshot shows a code editor with two files open: `input.txt` and `output.txt`. Both files are located in the directory `lab0 > Task 3`. The `input.txt` file contains the number `327305` on line 1. The `output.txt` file contains the number `5` on line 1. The editor interface includes a toolbar with icons for undo, redo, and other editing functions.

Микро-вывод

В этом задании мы вспомнили о том, как в Python можно вычислять остаток от деления, что заодно помогло нам узнать последнюю цифру искомого числа Фибоначчи.

Задание 4. Тестирование ваших алгоритмов

Задание – необходимо протестировать время выполнения вашего алгоритма в Задании 2 и Задании 3.

Листинг кода:

```
import time
import os

t_start = time.perf_counter()

f = open(os.path.join(os.path.dirname(__file__), "input.txt"), "r")
n = int(f.readline())

o = open(os.path.join(os.path.dirname(__file__), "output.txt"), "w")

if n ≤ 1:
    o.write(str(n))
else:
    a, b = 0, 1
    for _ in range(2, n + 1):
        a, b = b, a + b

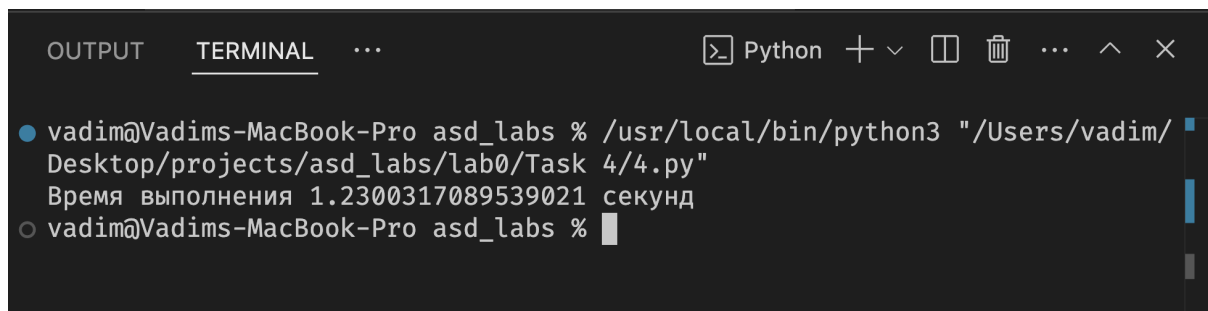
    o.write(str(b % 10))

f.close()
o.close()

print("Время выполнения %s секунд" % (time.perf_counter() - t_start))
```

Сначала импортируются необходимые модули `time` – для определения времени начала и конца работы скрипта, `os` – для определения пути к файлам `input.txt` и `output.txt`. Записываем в переменную время начала работы скрипта. Далее идет часть скрипта из задачи 3. После выполнения скрипта мы засекаем актуальное время, вычитаем из времени конца время начала и выводим на экран.

Результат работы кода:



```
OUTPUT  TERMINAL  ...
Python + - [] [X] ... ^ X

● vадim@Vadims-MacBook-Pro asd_labs % /usr/local/bin/python3 "/Users/vадim/Desktop/projects/asd_labs/lab0/Task 4/4.py"
  Время выполнения 1.2300317089539021 секунд
○ vадim@Vadims-MacBook-Pro asd_labs %
```

Микро-вывод

В этом задании мы узнали, как можно засекать время работы программы в Python.

Вывод

Таким образом, в ходе выполнения лабораторной работы были рассмотрены и реализованы различные задачи по теме работы с файлами и алгоритмов для вычисления чисел Фибоначчи. Была проведена работа с вводом-выводом данных, как через консоль, так и через файлы, что позволило вспомнить работы, как работать с языком Python.