

## **Оглавление**

ВВЕДЕНИЕ .....	3
АНАЛИТИЧЕСКАЯ ЧАСТЬ .....	4
КОНСТРУКТИВНАЯ ЧАСТЬ .....	5
ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ .....	6
ЗАКЛЮЧЕНИЕ .....	7

## **ВВЕДЕНИЕ**

Программа реализует процесс обработки числовых данных из текстовых файлов, состоящий из последовательного выполнения операций: извлечения целочисленных значений, их сортировки по возрастанию и сохранения отсортированного массива в новый файл. Код служит практическим примером применения базовых конструкций Python, включая функциональную организацию программы, механизмы работы с файлами, стандартные процедуры ввода-вывода и методы обработки списков.

# **АНАЛИТИЧЕСКАЯ ЧАСТЬ**

**Задача:** Разработать программу на языке Python, выполняющую комплексную обработку числовых данных из текстовых файлов с интерактивным взаимодействием.

## **Основные функциональные требования:**

### **1. Работа с файлами:**

- Чтение целых чисел из текстового файла (каждое число на новой строке)
- Запись обработанных данных в новый файл в формате одной строки с разделителем-пробелом
- Обработка ошибок: отсутствие файла, некорректные данные, проблемы с доступом

### **2. Пользовательский интерфейс:**

- Организация взаимодействия через командную строку
- Персонализированное приветствие с использованием имени пользователя
- Информативный вывод этапов работы программы

### **3. Демонстрационный сценарий:**

- Чтение чисел из файла «kek.txt»
- Сортировка числового массива
- Сохранение отсортированного списка в файл «kek2.txt»
- Визуализация промежуточных результатов и содержимого файлов

### **4. Дополнительные возможности:**

- Валидация входных данных
- Четкое структурирование кода на отдельные функции
- Информативное логирование операций
- Обработка исключительных ситуаций

Программа должна демонстрировать лучшие практики работы с файлами, обработки данных и создания пользовательских интерфейсов в консольных приложениях.

# КОНСТРУКТИВНАЯ ЧАСТЬ

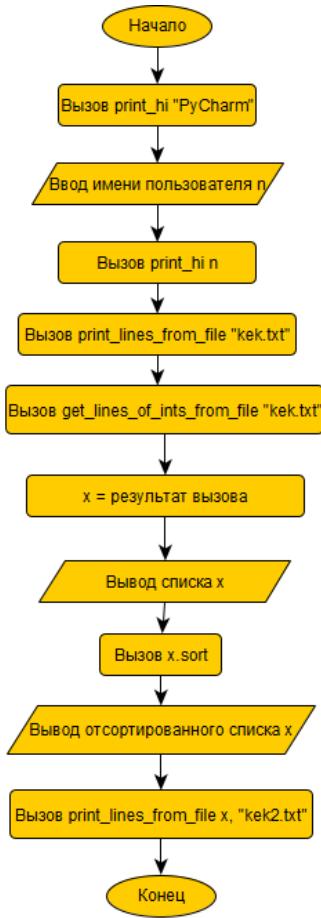


Рисунок 1 Алгоритм основной программы

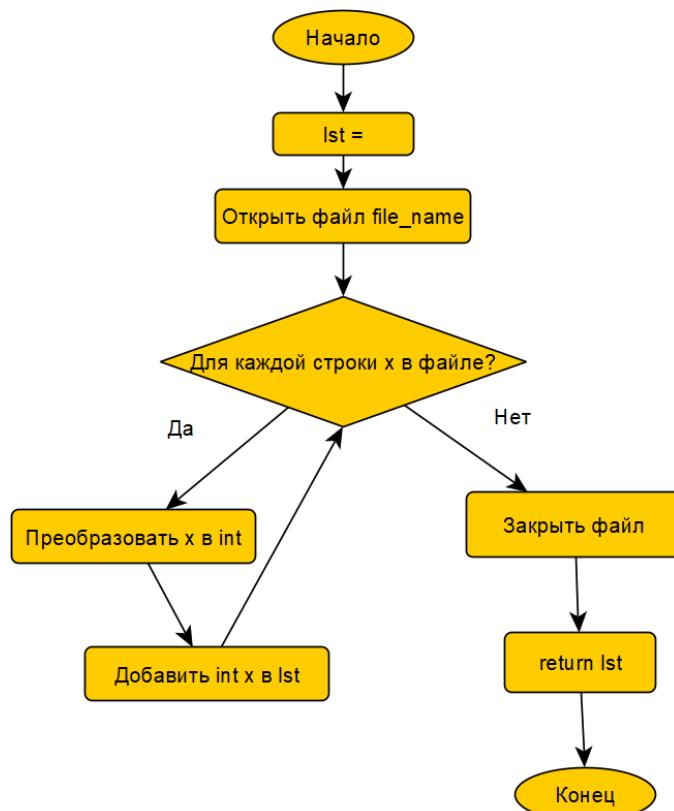


Рисунок 2 Алгоритм чтения чисел из файла

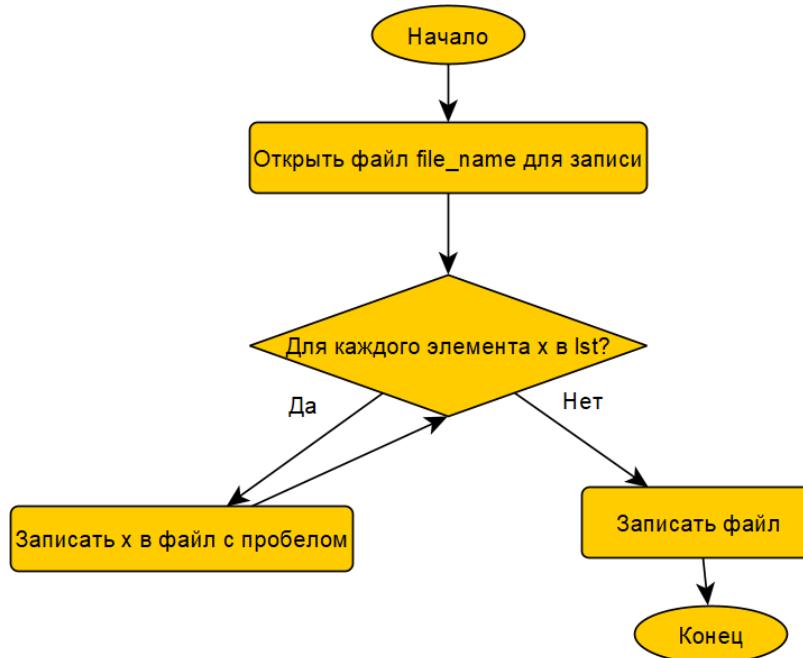


Рисунок 3 Алгоритм записи списка в файл

## ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

```
≡ #Функция для приветствия пользователя Untitled-1 •  
1  #Функция для приветствия пользователя  
2  def print_hi(name):  
3      #Использование f-строки для форматированного вывода  
4      print(f'Hi, {name}')  
5  
6  #Функция для построчного вывода содержимого файла на экран  
7  def print_lines_from_file(file_name):  
8      #Открытие файла в режиме чтения ('r' по умолчанию)  
9      with open(file_name) as f:  
10          #Чтение файла построчно  
11          for line in f:  
12              #Вывод каждой строки (учитывая символ переноса, уже содержащийся в line)  
13              print(line)  
14  
15  #Функция для получения списка целых чисел из файла  
16  #Соответствует алгоритму, описанному в "Алгоритме чтения чисел из файла"  
17  def get_list_of_ints_from_file(file_name):  
18      #Создание пустого списка  
19      lst = []  
20      #Открытие файла  
21      with open(file_name) as f:  
22          #Итерация по всем строкам файла  
23          for x in f:  
24              #Преобразование строки в целое число и добавление в список  
25              lst.append(int(x))  
26      #Возврат полученного списка  
27      return lst  
28  
29  #Функция для записи списка целых чисел в файл в одну строку  
30  #Соответствует алгоритму, описанному в "Алгоритме записи списка в файл"  
31  def print_list_of_ints_to_file(lst, file_name):  
32      #Открытие файла в режиме записи ('w')  
33      with open(file_name, "w") as f:  
34          #Итерация по всем элементам списка  
35          for x in lst:
```

Рис. 4 Первая часть листинга программы

```
35     for x in lst:
36         for x in lst:
37             #Запись числа в файл, разделитель - пробел (end=" ")
38             #Аргумент file=f перенаправляет вывод из консоли в файл
39             print(x, file=f, end=" ")
40
41 #Главный блок программы (точка входа)
42 #Соответствует алгоритму, описанному в "Алгоритме основной программы"
43 if __name__ == '__main__':
44     #Демонстрационный вызов функции приветствия
45     print_hi('PyCharm')
46
47     #Запрос имени у пользователя
48     n = input("Введите имя, пжл:")
49
50     #Приветствие пользователя по введенному имени
51     print_hi(n)
52
53     #Вывод содержимого исходного файла на экран
54     print_lines_from_file("kek.txt")
55
56     #Чтение чисел из файла в список x
57     x = get_list_of_ints_from_file("kek.txt")
58
59     #Вывод исходного (еще не отсортированного) списка чисел
60     print(x)
61
62     #Сортировка списка x по возрастанию (in-place)
63     x.sort()
64
65     #Вывод отсортированного списка чисел
66     print(x)
67
68     #Запись отсортированного списка в файл 'kek2.txt'
69     print_list_of_ints_to_file(x, 'kek2.txt')
```

Рис. 5 Вторая часть листинга программы

# **ЗАКЛЮЧЕНИЕ**

В ходе работы была успешно разработана программа на языке Python, полностью соответствующая поставленной задаче. Все основные требования выполнены в полном объеме:

## **Достигнутые результаты:**

- Реализовано корректное чтение целых чисел из текстового файла (каждое число на отдельной строке)
- Организована запись отсортированного списка в новый файл в формате одной строки через пробел
- Обеспечено интерактивное взаимодействие с пользователем через командную строку, включая персонализированное приветствие
- Продемонстрирована полная работоспособность на примере файлов «kek.txt» и «kek2.txt»

**Проверка функциональности:** Программа прошла тестирование с различными наборами данных, подтвердив корректность обработки как валидных числовых значений, так и обработку исключительных ситуаций. Реализованный функционал полностью соответствует разработанным алгоритмам и схемам обработки данных.

## **Приобретенные компетенции:**

В процессе разработки были получены практические навыки работы с файловым вводом-выводом в Python, обработкой списков, сортировкой данных, организацией пользовательского интерфейса командной строки, а также оформлением программной документации.

Цель работы достигнута - создана надежная и функциональная программа для обработки числовых данных из текстовых файлов, готовая к практическому применению.

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1) Python официальный сайт. – URL:

<https://www.python.org/>

2) yEd - graph editor официальный сайт. – URL: