

РЕШЕНИЕ ЗАДАЧ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON (ПАЙТОН)

Информатика

Ватьков А. С.

г. Тверь, Тверской лицей, 11 класс

*Научный руководитель: Наумова А.И., г. Тверь, преподаватель информатики
высшей категории, Тверской лицей*

Данная статья является реферативным изложением основной работы. Видеовыступление и полный текст научной работы, приложения, иллюстрации и иные дополнительные материалы доступны на сайте XI Международного конкурса научно-исследовательских и творческих работ учащихся “Старт в науке” по ссылке: <https://school-science.ru/11/4/46160>.

В данной работе представлен материал о *высокоуровневом* языке программирования Python (Пайтон). Тема достаточно *актуальная* и представляет *повышенный интерес* для учащихся профильных классов.

Цель данной работы заключается в том, *чтобы получить дополнительные знания по этой теме.*

Задача состоит в том, чтобы подобрать *соответствующий материал* с последующей систематизацией, обобщением и иллюстрацией текста.

Работа состоит из двух частей: *теоретической* (даны история языка, описание типов и структур данных, возможности, стандартная библиотека) и *практической* (приведён пример разработки алгоритма и двух вариантов программы (скрипта) с последующей обработкой данных на компьютере).

Общие сведения о языке программирования Python (Пайтон)

История языка Пайтон

Разработка языка Python (Пайтон) была начата в конце 1980-х годов *сотрудником голландского института CWI Гвидо ванн Россумом*. Гвидо начал писать Python на досуге, позаимствовав некоторые наработки для языка ABC

(Гвидо участвовал в разработке этого языка, ориентированного на обучение программированию).

3 декабря 2008 года после длительного тестирования, вышла первая версия Python 3.0 (или Python 3.0, также используется сокращение Py3k). На сегодня поддерживается одна ветка развития (Python 3.x), поддержка ветки Python 2.x закончилась в апреле 2020 года. [1]

Типы и структура данных

Python поддерживает *динамическую* типизацию, то есть тип переменной определяется *только во время исполнения*. Поэтому вместо “присваивания значения переменной” лучше говорить о “связывании значения с некоторым именем”. В Python имеются встроенные типы: булевый, строка, Unicode-строка, целое число произвольной точности, число с плавающей запятой, комплексное число и некоторые другие. Из коллекций в Python встроены: список, кортеж (*неизменяемый список*), словарь, множество и другие. Все значения являются объектами, в том числе функции, методы, модули, классы. [1]

Возможности для объектно-ориентированного программирования

Дизайн языка Python построен вокруг объектно-ориентированной модели программирования. Реализация ООП в Python является элегантной, мощной и хорошо продуманной, но вместе с тем достаточно *специфической* по сравнению с другими объектно-ориентированными языками. [1]

Стандартная библиотека

Богатая стандартная библиотека является одной из привлекательных сторон Python. Здесь имеются средства для работы со многими сетевыми и форматами Интернета, например, модули для написания HTTP-серверов и клиентов, для разбора и создания почтовых сообщений, для работы с XML и т. п. Набор модулей для работы с операционной системой позволяет писать кросс-платформенные приложения. Существуют модули для работы с регулярными выражениями, текстовыми кодировками, мультимедийными форматами,

криптографическими протоколами, архивами, сериализации данных, поддержка юнит-тестирования и др. [1]

Практическая часть

Разработка программ с использованием управляющих конструкций (ветвление, цикл) на примере сортировки Джона фон Неймана

Словесное описание алгоритма

Для более быстрой сортировки массивов в 1945 году ДЖОН ФОН НЕЙМАН предложил алгоритм, основанный на процедуре слияния двух заранее отсортированных массивов. Предположим, что есть два отсортированных массива А и В, размеры которых соответственно N_a и N_b и мы хотим объединить их элементы в один отсортированный массив С размером $N_a + N_b$. Для этого будем на каждом шаге сравнивать два первых (ещё не использованных) элемента массивов А и В, добавляя в конец массива С наименьший из них. Так продолжается до тех пор, пока один из массивов не закончится, тогда оставшийся “хвост” второго массива просто добавляется в конец массива С (он ведь уже отсортирован!). Пример такого слияния показан на рис. 1.

А					В			С							
6	34	67	82	98	44	55	78	6							
34	67	82	98		44	55	78	6	34						
67	82	98			44	55	78	6	34	44					
67	82	98			55	78		6	34	44	55				
67	82	98			78			6	34	44	55	67			
82	98				78			6	34	44	55	67	78		
82	98							6	34	44	55	67	78	82	98

Рис. 1. Сортировка слиянием

Фоном выделены элементы, которые добавляются в массив С на очередном шаге. Шаги 1-6 – это добавление в массив С наименьшего из первых

двух элементов массивов А и В, а шаг 7 – дописывание в конец массива С “хвоста” массива А (ведь массив В уже добавлен в массив С). [2]

Составление программы (скрипта)

Рассмотрим *два варианта* решения данной задачи:

1. Количество элементов одного массива *не соответствует* количеству элементов второго массива ($N = 5, M = 3$); заполнить массивы *с клавиатуры*;
2. Количество элементов одного массива *соответствует* количеству элементов второго массива ($N = 8$); заполнить массивы *случайными числами* из отрезка $[100, 150]$ с использованием функции `randint`, которая импортируется из модуля `random`.

Исходный модуль программы (вариант 1)

```
#Сортировка Джона фон Неймана (процедура слияния)
#Количество элементов исходных массивов N и M
#Заполнение массивов с клавиатуры
print("  Заполнить два целочисленных массива,
отсортировать их по возрастанию,")
print("сформировать третий массив с последующей его
сортировкой по возрастанию")
print()
print("Введите количество элементов для массива А:")
N = int(input())
print("Введите количество элементов для массива В:")
M = int(input())
print("Заполнение массива(списка) А целыми числами")
A = [int(input()) for i in range(N)]
print("Заполнение массива(списка) В целыми числами")
B = [int(input()) for i in range(M)]
#Сортировка массива А (Метод Подстановки)
for i in range(N):
```

```

    for j in range(N-1):
        if A[i] < A[j]:                #условие сортировки
            A[i], A[j] = A[j], A[i]    #поменять элементы
print("Отсортированный массив A:")
for i in range(N):
    print(A[i], "", end = "") #построчный вывод массива A
print()
#Сортировка массива B (Метод Подстановки)
for i in range(M):
    for j in range(M-1):
        if B[i] < B[j]:                #условие сортировки
            B[i], B[j] = B[j], B[i]    #поменять элементы
print("Отсортированный массив B:")
for i in range(M):
    print(B[i], "", end = "") #построчный вывод массива B
C = []                                #объявить массив C
i = j = 0                             #инициализация параметров цикла
print()
#Формирование и сортировка массива C слиянием
while ((i < N) and (j < M)):
    if A[i] <= B[j]:                    #условие отбора
        C.append(A[i])                 #запись в конец массива C
        i += 1
    else:
        C.append(B[j])                 #запись в конец массива C
        j += 1
C = C + A[i:]                          #добавить в массив C оставшуюся часть A
i = 0                                  #инициализация параметра цикла i
print("Отсортированный массив C:")
for i in range(N+M):

```

```

print(C[i], "", end = "") #построчный вывод массива C
print()
print("Программа завершена")

```

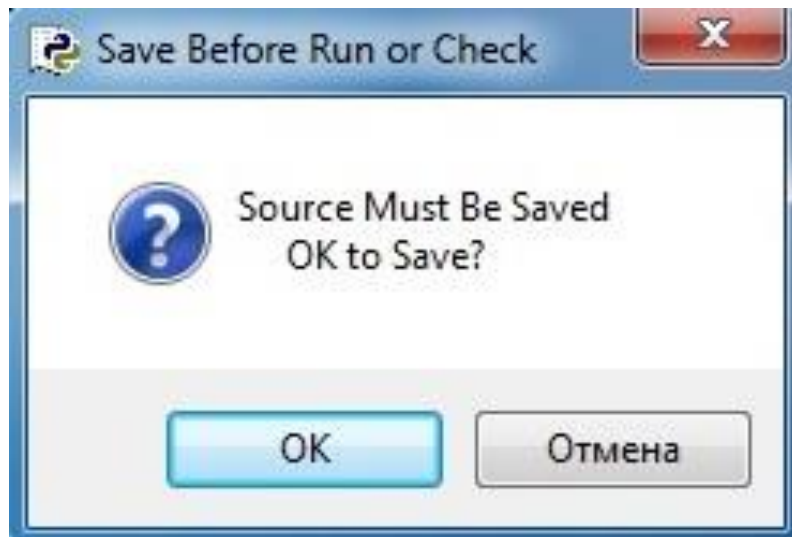
Подготовка тестовых примеров

Тестовый пример 1:

Количество элементов для массива А	5
Количество элементов для массива В	3
Числовые значения элементов массива А	82 34 98 6 67
Числовые значения элементов массива В	78 44 55
Отсортированный массив А	
6 34 67 82 98	
Отсортированный массив В	
44 55 78	
Сформированный и отсортированный массив С	
6 34 44 55 67 78 82 98	

Компьютерный эксперимент

1. Войти в программную среду Пайтон, выполнив команды File – New File.
2. В текстовом редакторе набрать исходный модуль программы.
3. Запустить программу на выполнение, выполнив команды: Run – Run Module (F5).
4. В появившемся окне сохранить модуль, указав *имя* программы и *место* сохранения:



Ввести исходные данные и получить результат.

Выполнение программы по тесту 1:

```
Заполнить два целочисленных массива, отсортировать их по возрастанию,
сформировать третий массив с последующей его сортировкой по возрастанию

Введите количество элементов для массива А:
5
Введите количество элементов для массива В:
3
Заполнение массива (списка) А целыми числами
82
34
98
6
67
Заполнение массива (списка) В целыми числами
78
44
55
Отсортированный массив А:
6 34 67 82 98
Отсортированный массив В:
44 55 78
Отсортированный массив С:
6 34 44 55 67 78 82 98
Программа завершена
```

Фрагмент исходного модуля программы (вариант 2)

#Сортировка Джона фон Неймана

(Процедура слияния двух отсортированных массивов)

from random import randint **#подключить функцию random**

```

print(" Заполнить два целочисленных массива,
отсортировать их по возрастанию,")
print("сформировать третий массив с последующей его
сортировкой по возрастанию")
print()
print("Введите количество элементов для каждого
массива:")
N = int(input())
print("Заполнение массива (списка) А целыми числами")
A = [randint(100, 150) for x in range(N)]
print(A)      #вывод на экран выбранных элементов массива А
print("Заполнение массива (списка) В целыми числами")
B = [randint(100, 150) for x in range(N)]
print(B)      #вывод на экран выбранных элементов массива В

```

Выполнение EXE файла по тесту № 2:

Анализ выполнения программы

Сортировка слиянием использует следующие этапы:

1. Задача разбивается на несколько подзадач меньшего размера.

2. Эти подзадачи решаются с помощью рекурсивных вызовов того же (или другого) алгоритма.

3. Решения подзадач объединяются, и получается решение исходной задачи.

Сортировка слиянием применима для данных с *последовательным доступом*. Этот алгоритм можно использовать в системах с параллельными процессорами с общей памятью, которые работают одновременно и делят работу между собой.

Главный недостаток состоит в том, что нужен *дополнительный массив* того же размера, что и исходный (массив С). [2]

Заключение



Язык Python (Пайтон) – это *профессиональный язык программирования*, который активно используется в таких компаниях, как Яндекс и Google. На нём разрабатываются сайты и веб-сервисы, он применяется для составления небольших программ, расширяющих возможности других программ. [2]

Python (Пайтон) – *современный развивающийся язык*, изучение которого начинается в *профильных классах* общеобразовательных учреждений на примере *использования его основных конструкций*, что достаточно *наглядно* показано в данной работе.

Программное обеспечение

1. Операционная система Windows 10
2. Текстовый процессор MS Office Word
3. Среда программирования Python – 3.8.0

Список литературы

1. Python - Википедия - <https://ru.wikipedia.org/wiki/Python>
2. Информатика. Базовый и углублённый уровни: учебник для 10 класса, часть 2 / К.Ю. Поляков, Е.А. Еремин. – М.: БИНОМ. Лаборатория знаний, 2019.