

**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**(РУТ (МИИТ)**

|  |
| --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Кафедра «Управление и защита информации»

**Отчет по заданию**

«Реализация алгоритма сортировки вставкой»

**по дисциплине**

«Методы программирования»

**Выполнил:** студент

группы ТКИ-311

Станчук П.Н.

**Проверил:** доценты кафедры УиЗИ, к.т.н. Логинова Л. Н. и

Сафронов А.И.

**Москва 2022 г.**

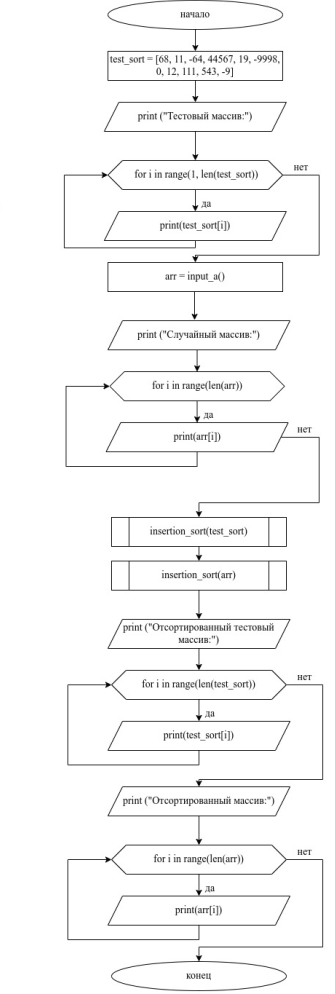
**1. Цель работы**

Изучение алгоритма сортировки вставкой.

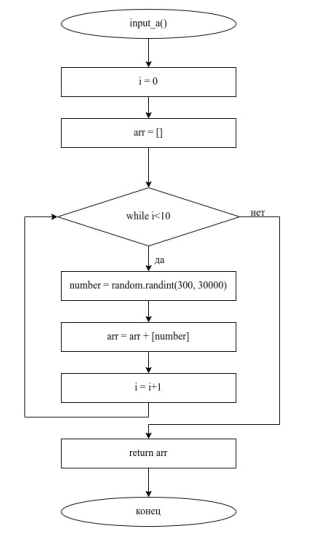
**2. Формулировка задачи**

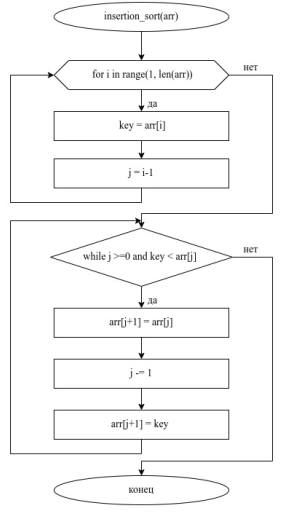
Реализовать программу, выполняющую алгоритм сортировки вставкой.

**3. Блок-схема алгоритма**

****



****

****



**4. Подбор тестовых примеров**



Ввод: массив из 10 элементов: 68, 11, -64, 44567, 19, -9998, 0, 12, 111, 543.



Вывод: -9998, -64, 0, 11, 12, 19, 68, 111, 543, 44567.



**5. Код программы:**

import time

startTime1 = time.time() # время начала замера

def insertion\_sort(arr):

for i in range(1, len(arr)):

key = arr[i]

j = i-1

while j >=0 and key < arr[j] :

arr[j+1] = arr[j]

j -= 1

arr[j+1] = key

def input\_a():

import random

i = 0

arr = []

while i<20:

number = random.randint(300, 30000)

arr = arr + [number]

i = i+1

return arr

test\_sort = [68, 11, -64, 44567, 19, -9998, 0, 12, 111, 543]

print ("Тестовый массив:")

for i in range(len(test\_sort)):

print(test\_sort[i])

arr = input\_a()

print ("Отсортированный тестовый массив:")

for i in range(len(test\_sort)):

print(test\_sort[i])

print ("Случайный массив:")

for i in range(len(arr)):

print(arr[i])

insertion\_sort(test\_sort)

insertion\_sort(arr)

print ("Отсортированный массив:")

for i in range(len(arr)):

print(arr[i])

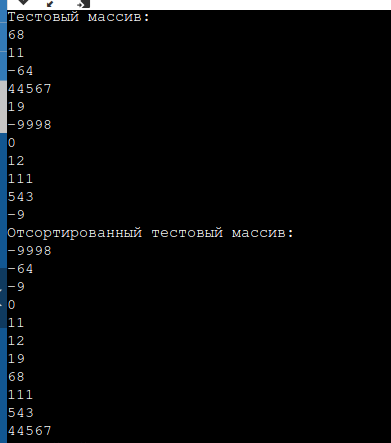
endTime1 = time.time() #время конца замера

totalTime1 = endTime1 - startTime1 #вычисляем затраченное время

print ("Время, затраченное на сортировку:", totalTime1, "секунд")

**6. Результат расчёта тестовых примеров на ПК**

**Проверим работоспособность алгоритма**



Тестовый массив отсортирован верно.

Сравним время работы алгоритма. Для этого закомментируем обработку тестового массива.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

На сортировку массива из 20 элементов затрачено 0.0012743473052978516 секунд.

На сортировку массива из 100 элементов затрачено 0.010003805160522461 секунд, то есть при увеличении количество элементов в 5 раз время увеличилось приблизительно в 8 раз.



**7. Вывод**

В данной работе я реализовала алгоритм сортировки вставкой на Python и протестировала его, а также узнала, что при увеличении количества элементов массива в 5 раз время сортировки вставкой увеличилось приблизительно в 8 раз.