|  |
| --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  «Национальный исследовательский ядерный университет «МИФИ» |
| **Димитровградский инженерно-технологический институт –**  филиал федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский ядерный университет «МИФИ»  **(ДИТИ НИЯУ МИФИ)** |

**Кафедра информационных технологий**

**ОТЧЕТ**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

Тема: Разработка программного продукта «Компьютерная безопасность»

студента группы ИСТ-11

Колесников М. П.

(фамилия, инициалы)

Руководитель от кафедры: Павлова Т.Н.

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Димитровград, 2022

СОДЕРЖАНИЕ

[**ВВЕДЕНИЕ 3**](#_Toc109310374)

[1. РАЗРАБОТКА АЛГОРИТМА ПРОГРАММНОГО ПРОДУКТА ОПРЕДЕЛЕНИЯ СХОЖЕСТИ НОМЕРОВ И ПИН-КОДОВ 3](#_Toc109310375)

[1.1. Основные положения 3](#_Toc109310376)

[1.2. Выбор языка программирования 4](#_Toc109310377)

[1.3. Разработка алгоритма для определения схожести номеров и пин-кодов 5](#_Toc109310378)

[2. ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА 8](#_Toc109310379)

[ЗАКЛЮЧЕНИЕ 10](#_Toc109310380)

[Приложение А 11](#_Toc109310381)

ВВЕДЕНИЕ

В наше время информационной безопасности уделяется особое внимание. Многие люди хранят свои личные данные в интернете и не подозревают, что они в любой момент могут быть украдены. Случается такое, что виновными в утечке данных оказываются не взломщики, а некомпетентные сотрудники или ошибки в алгоритмах. На сегодняшний день развитие информационной безопасности — одна из важнейших задач в мире информационных технологий.

По данной учебной практике мной была выбрана тема: "Компьютерная безопасность".

Целью данной работы является разработка алгоритма для выявления похожих номеров и пин-кодов.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. Выбрать язык программирования для реализации алгоритма в виде программного кода.
2. Разработать алгоритм для определения схожести номеров и пин-кодов.
3. Применить разработанный алгоритм в написании программного кода на выбранном языке программирования.
4. Выполнить тестирование полученного программного продукта.

Объектом исследования является алгоритм для схожести номеров и пин-кодов.

Предметом исследования является программная реализация алгоритма для определения схожести номеров и пин-кодов.

# 1. РАЗРАБОТКА АЛГОРИТМА ПРОГРАММНОГО ПРОДУКТА ОПРЕДЕЛЕНИЯ СХОЖЕСТИ НОМЕРОВ И ПИН-КОДОВ

## 1.1. Основные положения

Похожими считаются два номера, если один из них можно получить из другого вставкой, удалением или заменой одной цифры. Ведущих нулей в записи номера быть не может. Нумерация происходит последовательными целыми числами, начиная с единицы, а пин-коды всех представляют собой строку из четырёх шестнадцатеричных цифр.

## 1.2. Выбор языка программирования

Язык программирования — формальный язык, предназначенный для записи компьютерных программ. Язык программирования определяет набор лексических, синтаксических и семантических правил, определяющих внешний вид программы и действия, которые выполнит исполнитель (обычно — ЭВМ) под её управлением.

Python — высокоуровневый язык программирования общего назначения. Он имеет минималистичный синтаксис и направлен на повышение читаемости кода (и в целом на повышение производительности разработчика).

Плюсами Python является:

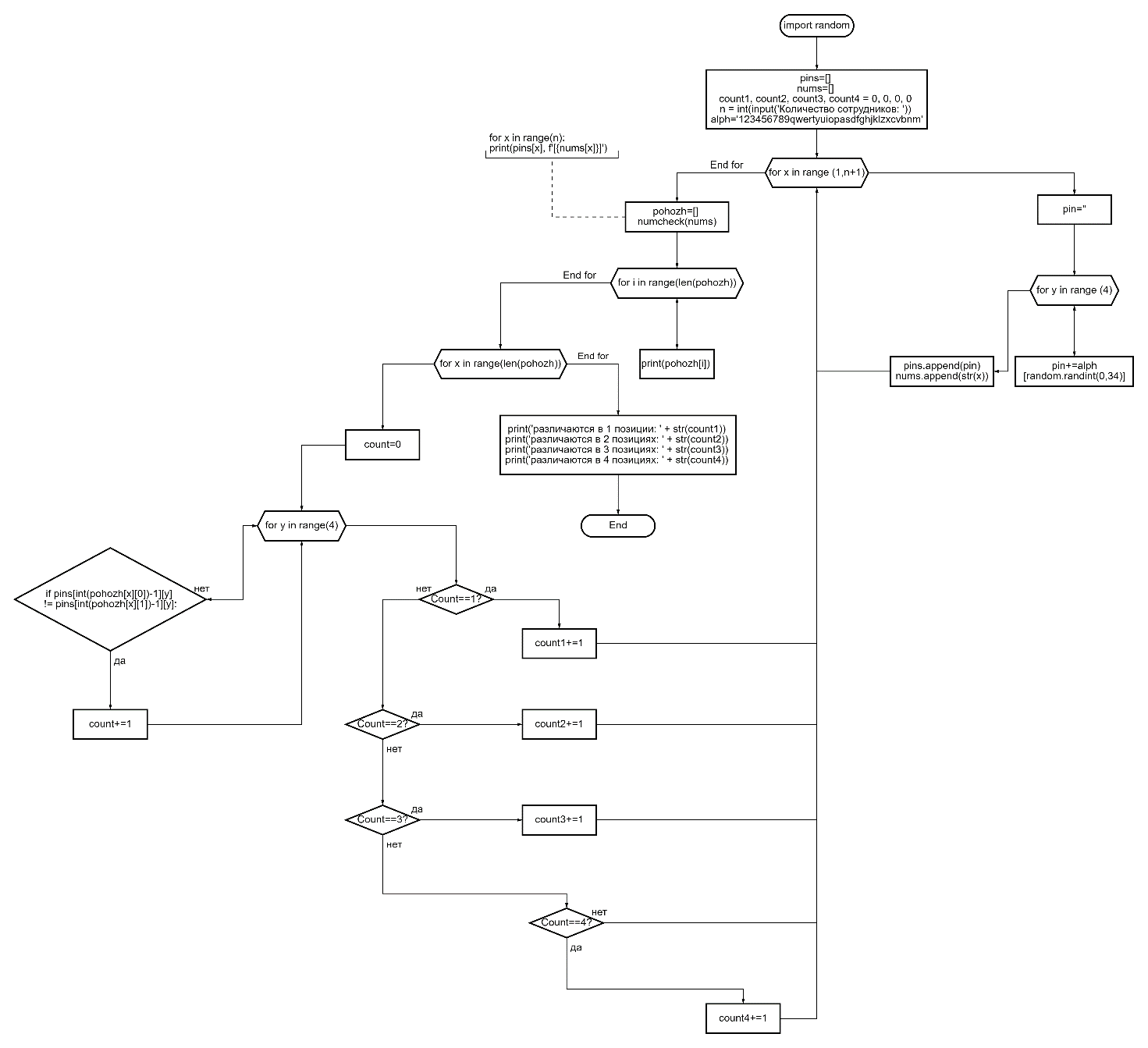
1. Низкий порог вхождения;
2. Язык широкого применения;
3. Минималистичный синтаксис;
4. Кроссплатформенность;
5. Открытый исходный код интерпретатора Python;
6. Наличие дружелюбного, отзывчивого сообщества;
7. Поддержка многих IDE;
8. Огромное количество библиотек;
9. Входит в поставку большинства дистрибутивов Linux.

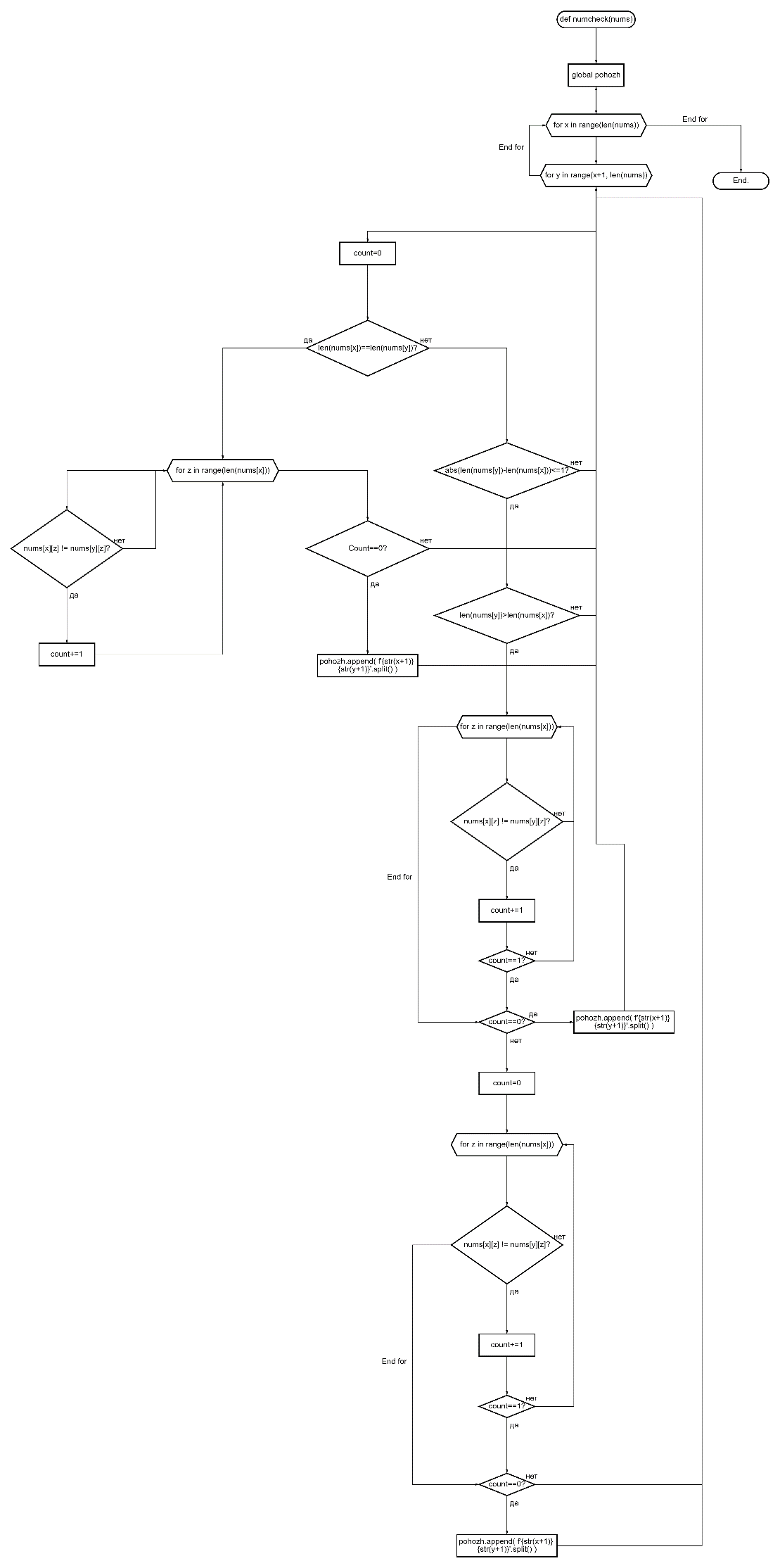
Также у языка программирования Python есть и минусы:

1. Низкая производительность
2. Отсутствие статический типизации

Было принято решение использовать Python как основной язык программирования, т.к. он сочетает в себе простоту и удобство, а его минусы имеют минимальное значение относительно данного продукта, т.к. его объём не велик и потеря производительности по сравнению с другими, более быстрыми, языками программирования незаметна для пользователя

## 1.3. Разработка алгоритма для определения схожести номеров и пин-кодов





Пользователю необходимо ввести число N сотрудников, далее для N сотрудников генерируются пин-коды и происходит запись пин-кодов и личных номеров пользователей в два разных массива таким образом, что индексы личных номеров и пин-кодов совпадают. Затем отбираются пары пользователей с похожими личными номерами. Пин-коды пользователей отобранных пар сравниваются на схожесть, результаты записываются в переменные count1, count2, count3, count4, и когда программа завершается, эти переменные выводятся на экран.

# 2. ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА

Тестирование показало, что во всех случаях программа работает исправно, но чем больше вводимое число N, тем медленнее работает программа. Это обусловлено тем, что Python – ЯП с невысокой скоростью воспроизведения кода.

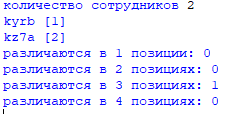


Рисунок 2.1 — Тест 1

При небольших входных данных программа работает быстро. При входных данных> 2000 программа начинает работать ощутимо медленнее.

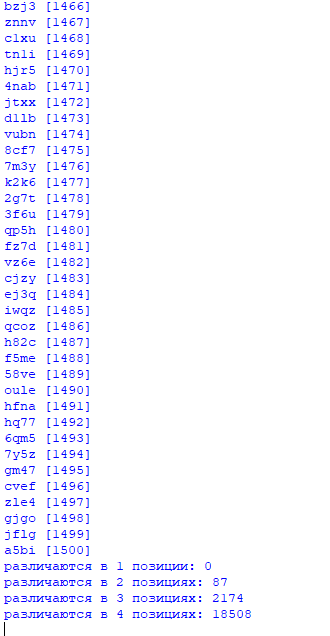


Рисунок 2.2 — Тест 2

В работе программы исключены ошибки, т. к. код полностью продуман, и всё, что требуется от пользователя — подать на вход программы число.

# ЗАКЛЮЧЕНИЕ

В процессе прохождения учебной практики были выполнены все поставленные задачи, был спроектирован и реализован программный продукт, позволяющий сравнивать схожесть номеров и четырёхзначных пин-кодов. Для реализации был выбран и изучен язык программирования Python.

# Приложение А

Код программного продукта:

import random

pins=[]

nums=[]

count1, count2, count3, count4 = 0, 0, 0, 0

n = int(input('количество сотрудников '))

alph='123456789qwertyuiopasdfghjklzxcvbnm'

for x in range(1,n+1):

pin=''

for y in range(4):

pin+=alph[random.randint(0,34)]

pins.append(pin)

nums.append(str(x))

for x in range(n):

print(pins[x], f'[{nums[x]}]')

pohozh=[]

def numcheck(nums):

global pohozh

for x in range(len(nums)):

for y in range(x+1, len(nums)):

count=0

if len(nums[x])==len(nums[y]):

for z in range(len(nums[x])):

if nums[x][z] != nums[y][z]:

count+=1

if count == 1:

pohozh.append( f'{str(x+1)} {str(y+1)}'.split() )

elif abs(len(nums[y])-len(nums[x]))<=1:

if len(nums[y])>len(nums[x]):

for z in range(len(nums[x])):

if nums[x][z] != nums[y][z]:

count+=1

if count==1:

break

if count==0:

pohozh.append( f'{str(x+1)} {str(y+1)}'.split() )

else:

count=0

for z in range(len(nums[x])):

if nums[x][z] != nums[y][z+1]:

count+=1

if count==1:

break

if count==0:

pohozh.append( f'{str(x+1)} {str(y+1)}'.split() )

numcheck(nums)

#for i in range(len(pohozh)):

#print(pohozh[i])

for x in range(len(pohozh)):

count=0

for y in range(4):

if pins[int(pohozh[x][0])-1][y] != pins[int(pohozh[x][1])-1][y]:

count+=1

if count==1:

count1+=1

elif count==2:

count2+=1

elif count==3:

count3+=1

elif count==4:

count4+=1

print('различаются в 1 позиции: ' + str(count1))

print('различаются в 2 позициях: ' + str(count2))

print('различаются в 3 позициях: ' + str(count3))

print('различаются в 4 позициях: ' + str(count4))