МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК

Практическое занятие № 16.

ФАЛОВЫЙ ТИП ДАННЫХ.

Вариант 3.

Выполнил студент:

Герасимов Константин Сергеевич

ПиОА-01.02

**1. Решение задачи 5.1**

***1.1 Постановка задачи***

Текстовый файл может содержать данные следующего типа записанные через соответствующий разделить:

* название пункта назначения рейса;
* номер рейса;
* тип самолета.

Написать программу, выполняющую следующие действия:

* ввод с клавиатуры данных и запись их в файл

**Дано:**

**Функции:**

def add\_info(flightname: str, flightnum: str, typeofairplane: str): - функция записи информации в файл.

**Входные параметры:**

**Flightname = input() – место назначения**

**Flightnum = input() – номер рейса**

**Typeofairplanе = input() – тип самолёта**

**Выходные параметры:**

f.write(f'{flightname};{flightnum};{typeofairplane}' + '\n') – запись в файл информации.

***1.2 Постановка задачи***

Текстовый файл может содержать данные следующего типа записанные через соответствующий разделить:

* название пункта назначения рейса;
* номер рейса;
* тип самолета.

Написать программу, выполняющую следующие действия:

* Вывод на экран информации о рейсе, номер которого введен с клавиатуры, если таких рейсов нет, вывести соответствующее сообщение.

**Дано:**

**Функции:**

def view\_info(flightnum: str): - функция для вывода данных из файла на экран.

def show\_me\_db(start\_ind: int, end\_ind: int): - вывод базы данных в диапазоне.

**Входные параметры:**

**Flightname = input() – место назначения**

**Flightnum = input() – номер рейса**

**Typeofairplanе = input() – тип самолёта**

**Data\_base.txt – файл, база данных**

**Start\_ind = int(input()) – с какой строки начать печать базы данных**

**End\_ind = int(input()) – на какой строке закончить печать базы данных**

**Выходные параметры:**

print(f'Обнаружено {len(ready)} рейсов {flightnum}, выполняющихся в разное время: ')

print(f'Рейс {flightnum} следующий в город {ready[i][0].capitalize()} выполняется на борту {ready[i][2].capitalize()}', end='') – в случае если надено несколько рейсов с заданным номером.

print(f'Обнаружен {len(ready)} рейс {flightnum}: ')

print(f'Рейс {flightnum} следующий в город {ready[0][0].capitalize()} выполняется на борту {ready[0][2].capitalize()}', end='') – в случае если найден один рейс с заданным номером

print(f'Рейс с номером {flightnum} не обнаружен') – в случае если рейсы не обнаружены

print(flights[i], end='') – вывод базы данных в диапазоне

***1.3 Постановка задачи***

Организовать генерацию данных в файл не менее 1000 записей (необязательное дополнительное задание, оценивается дополнительным баллом).

**Дано:**

**Функции:**

def generate\_db(): - создание базы данных из 1000 записей.

**Входные параметры:**

Data\_base.txt – файл, база данных

**Начальные значения:**

cities = ['astana', 'moscow', 'london', 'Saint-Peterburg', 'tokyo', 'toronto', 'atlanta', 'krasnoyrarsk', 'rostov-on-Don', 'Vladivostok', 'kerch', 'Tyumen', 'artem', 'stockholm', 'budapesht', 'berlin'] – список городов полёта

airplanes = ['A320', 'Boeing770', 'Boeing707', 'SukhoiSuperJet'] – типы самолётов

**Выходные параметры:**

f.write(f'{cities[randint(0, len(cities)-1)]};{randint(100,999)};{airplanes[randint(0, len(airplanes)-1)]}' + '\n') – запись 1000 строк в файл

***1.3 Текст программы на языке Python***

from random import randint

def add\_info(flightname: str, flightnum: str, typeofairplane: str):

with open('Data\_base.txt', 'a') as f:

f.write(f'{flightname};{flightnum};{typeofairplane}' + '\n')

def view\_info(flightnum: str):

flightlist = []

ready = []

with open('Data\_base.txt', 'r') as f:

flights = f.readlines()

for num in range(len(flights)):

flightlist.append(flights[num].split(';'))

for i in range(0, len(flightlist)):

if flightnum == flightlist[i][1]:

ready.append(flightlist[i])

if ready != []:

if len(ready) > 1:

print(f'Обнаружено {len(ready)} рейсов {flightnum}, выполняющихся в разное время: ')

for i in range(len(ready)):

print(f'Рейс {flightnum} следующий в город {ready[i][0].capitalize()} выполняется на борту {ready[i][2].capitalize()}', end='')

elif len(ready) == 1:

print(f'Обнаружен {len(ready)} рейс {flightnum}: ')

print(f'Рейс {flightnum} следующий в город {ready[0][0].capitalize()} выполняется на борту {ready[0][2].capitalize()}', end='')

else:

print(f'Рейс с номером {flightnum} не обнаружен')

def generate\_db():

cities = ['astana', 'moscow', 'london', 'Saint-Peterburg', 'tokyo', 'toronto', 'atlanta', 'krasnoyrarsk', 'rostov-on-Don', 'Vladivostok', 'kerch', 'Tyumen',\

'artem', 'stockholm', 'budapesht', 'berlin']

airplanes = ['A320', 'Boeing770', 'Boeing707', 'SukhoiSuperJet']

with open('Data\_base.txt', 'a') as f:

for i in range(1000):

f.write(f'{cities[randint(0, len(cities)-1)]};{randint(100,999)};{airplanes[randint(0, len(airplanes)-1)]}' + '\n')

def show\_me\_db(start\_ind: int, end\_ind: int):

with open('Data\_base.txt', 'r') as f:

flights = f.readlines()

for i in range(start\_ind-1, end\_ind):

print(flights[i], end='')

def main():

while True:

print('Выберите пункт меню:\n1.Добавить информацию о рейсе;\n2.Вывести информацию о рейсе по его номеру;\n3.Сгенерировать базу данных из 1000 строк;\n4.Вывести кусок бaзы данных на экран;\n5.Выход.\nОтвет: ')

answer = input()

if answer == '1':

add\_info(input('Введите, пожалуйста, место назначения на английском: '),\

input('Введите, пожалуйста номер рейса: '), input('Введите тип самолёта: '))

continue

if answer == '2':

view\_info(input('Введите, пожалуйста, номер рейса, о котором вы хотите узнать информацию: '))

continue

if answer == '3':

generate\_db()

continue

if answer == '4':

show\_me\_db(int(input('Введите строку, с которой начнется вывод информации: ')), int(input('Введите строку, на которой вы хотите закончить вывод информации:')))

continue

if answer == '5':

break

main()

***1.3 Результат тестирования программы на языке Python***

Изображение выглядит как текст

Автоматически созданное описание

**2. Решение задачи 5.2**

***1.1 Постановка задачи***

Написать программу, которая считывает текст из файла и выводит на экран только строки, содержащие двузначные числа.

**Дано:**

**Функции:**

def udalitel(): - функция, удаляющая подстроки, не содержащие двузначные числа

**Начальные значения:**

nums = list(map(str, range(0, 10))) – список цифр

wre = [] – список, в который записывается итоговая строка

**Выходные параметры:**

print(wre[i], end=' ')

print() – строки, содержащие двузначные числа

***1.2 Текст программы на языке Python***

def udalitel():

nums = list(map(str, range(0, 10)))

wre = []

with open('file.txt', 'r') as f:

strokes = f.readlines()

for i in range(len(strokes)):

for sym in range(len(strokes[i])-2):

if (strokes[i][sym] in nums) and (strokes[i][sym+1] in nums) and (strokes[i][sym+2] not in nums):

wre.append(strokes[i])

for i in range(len(wre)):

print(wre[i], end=' ')

print()

udalitel()

***1.3 Результат тестирования программы на языке Python***

**Изображение выглядит как текст

Автоматически созданное описание**