**Федеральное государственное автономное образовательное учреждение**

**высшего образования**

**"Национальный исследовательский университет**

**"Высшая школа экономики"**

Московский институт электроники и математики им. А.Н. Тихонова НИУ ВШЭ

Департамент компьютерной инженерии

**Курс: Проектный семинар «Python о науке данных»**

Виртуальный кино-помощник

**Руководство пользователя**



**МОСКВА 2022**

**Оглавление**

Технические требования……………………………………………………………..3

Библиотеки……………………………………………………………………………4

Архитектура приложения……………………………………………………………5

Структура каталогов………………………………………………………………...11

Возможные модернизации………………………………………………………….12

Листинг модулей…………………………………………………………………….13

**Технические требования**

Для корректной работы приложения требуется пакет Anaconda 3, а персональный компьютер пользователя должен удовлетворять следующим минимальным системным требованиям:

1. Windows - 64-битная x86, 32-битная x86; MacOS - 64-битная x86; Linux - 64-битная x86, 64-битная Power8 / Power9.
2. RAM: 3+ GB
3. CPU: 2+ cores

А также иметь:

1. Оперативную память: 64 MB ОЗУ
2. Видеокарту: Any.
3. Место на диске: 10 MB.
4. Звуковую карту: N/A.

**Библиотеки**

В данном приложении были использованы следующие библиотеки:

1. tkinter
2. tkinter.ttk
3. os
4. numpy
5. pandas
6. pickle
7. PIL

А также:

1. from tkinter import \*
2. from tkinter import filedialog
3. from tkinter import messagebox as mb

**Архитектура приложения**

#Библиотека, содержащая универсальные модули:

checking - проверка значений

files - работа с файлом

save\_figure - сохранение фигуры matplotlib

from library import checking

from library import files

from library import save\_figure

\_\_all\_\_ = ["checking", "files", "save\_figure"]

#Модуль с функциями проверки вводимых значений

def checking\_number(string):

"""

Функция для проверки вводимого пользователем значения на

целое положительное число

Входные параметры:

строка

Выходные параметры:

True/False

"""

return str.isdecimal(str(string))

def checking\_positive(string):

"""

Функция для проверки вводимого пользователем значения на

положительное число

Входные параметры:

строка

Выходные параметры:

True/False

"""

try:

n=int(string)

if(n>0):

return True

else:

return False

except ValueError:

return False

def checking\_string(string):

"""

Функция для проверки вводимого пользователем значения на строку

только из символов

Входные параметры:

строка

Выходные параметры:

True/False

"""

if (not string.isalpha()):

return False

else:

return True

def checking\_empty\_string(string):

"""

Функция для проверки вводимого пользователем значения на

непустую строку

Входные параметры:

строка

Выходные параметры:

True/False

"""

if all(x.isspace() for x in string):

return False

else:

return True

def checking\_conditions(n,lim1=1000000,lim2=0):

"""

Функция для проверки числа на ограничения

Входные параметры:

число

число огр. сверху

число огр. снизу

Выходные параметры:

True/False

"""

if(n<=lim1 and n>=lim2):

return True

else:

return False

"""

Модуль с функциями для работы с файлами

"""

import pickle

def filename(path,name):

"""

Функция для создания названия файла с полным путем

Входные параметры:

строка - название файла

строка - путь

Выходные параметры:

строка - путь + название файла

"""

path\_name=path+'\\'+name

return path\_name

def save\_table(path\_name,typ,table):

"""

Функция для сохранения файла

Входные параметры:

строка - путь + название файла

тип файла - excel/pickle

таблица

Выходные параметры:

True/False

"""

try:

if(typ=='excel' or typ=='base'):

table.to\_excel(path\_name, index=False)

elif(typ=='pickle') :

with open(path\_name, 'wb') as f:

pickle.dump(table, f)

return True

except PermissionError:

return False

"""

Модуль с функцией сохранения фигуры matplotlib

"""

import matplotlib.pyplot as plt

import files

def save(figure,path,name):

"""

Функция сохранения графика в файл

Входные параметры:

Фигура

Название файла

Путь

Формат

Выходные параметры:

-

"""

figure.savefig(files.filename(path,name))

**Структура каталогов**

Информационно–аналитическое приложение размещается в стандартной структуре каталогов:

Work <- основной каталог.

Data — база данных.

Graphics — содержит копии графических отчетов.

Library — содержит библиотеку стандартных (универсальных) функций, разработанных бригадой, которые могут использоваться для создания других приложений.

Notes — документация.

Output — копии текстовых отчетов.

Scripts — содержит специализированный модуль и файл с определением параметров настройки приложения.

**Возможные модернизации**

Приложение принимает записи из базы данных, сформированной вручную, поэтому одной из модернизаций может быть формирование БД путем отправления и получения API-запросов к данным, находящимся в открытом доступе. Такой способ позволит сделать приложение, охватывающее огромное количество кинокартин, вследствие чего оно оно будет практически во всем превосходить существующие подборки и сортировки онлайн-платформ.

При таком объеме данных появится необходимость расширять список критериев для более детального и конкретного анализа данных.

**Листинг модулей**

Модуль настроек:

# -\*- coding: utf-8 -\*-

import os

text = os.path.abspath(os.path.join(os.getcwd(), '..', 'Output'))

graph = os.path.abspath(os.path.join(os.getcwd(), '..', 'Graphics'))

data = os.path.abspath(os.path.join(os.getcwd(), '..', 'Data'))

work =os.path.abspath(os.path.join(os.getcwd(), '..', ''))

types\_data = "excel"

types\_file= "excel"

base\_path=data+'\\'+types\_data

font = "Calibri"

'''Цвет фона'''

color\_back = "LightBlue2"

'''Цвет фона кнопок'''

color = "SteelBlue1"

'''Цвет текста'''

color\_text = "Blue4"

Листинг главного скрипта - файл.