ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

МОСКОВСКИЙ ИНСТИТУТ ЭЛЕКТРОНИКИ И МАТЕМАТИКИ

**РУКОВОДСТВО РАЗРАБОТЧИКА**

**для самостоятельной работы по индивидуальному плану**

**по курсу**

**«Проектный семинар “Python в науке о данных”»**

Студента группы БИВ224

Миннегалиевой Радмилы Рамисовны, [rrminnegalieva@edu.hse.ru](mailto:rrminnegalieva@edu.hse.ru), +79649600763

Тема работы:

Визуализация коронавируса с помощью тепловой карты и графиков

Руководитель:

Полякова Марина Васильевна

|  |  |
| --- | --- |
| «\_\_\_» \_\_\_\_\_\_\_\_\_ 202\_ г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись руководителя |

Задание выдано

|  |  |
| --- | --- |
| «\_\_\_» \_\_\_\_\_\_\_\_\_ 202\_ г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись руководителя |

Задание принято к исполнению

Содержание

1. Назначение и условия применения программы2
2. Характеристики и архитектура кода 3
   1. Используемые методы3
   2. Структура программы3
3. Входные и выходные данные6
   1. Входные данные6
   2. Выходные данные6

**Назначение и условия применения программы**

Программа представляет собой пользовательский интерфейс (десктопное приложение), где пользователь с помощью полей ввода (поля вида дат и выпадающего списка) для фильтрации информации предоставляет визуализированную графическую информацию. Программа предоставляет интерактивную карту и автоматически сгенерированные графики линейной зависимости процента загруженности аэропорта от отрезка дат карантинного времени.

Условия по характеристикам персонального компьютера, выполнение которых необходимо для генерации кода:

1. Процессор с тактовой частотой не менее 1 ГГц
2. Оперативную память объёмом не менее 4 Гб
3. Видеокарту, монитор, мышь, клавиатуру

Программа написана на языке программирования Python. Среда разработки, компилятор – Anaconda 2023.03

**Характеристика и структура кода**

**Используемые методы**

При разработке десктопного приложения использовались следующие библиотеки языка программирования Python:

1. dash 2.9.3
2. dash-bootstrap-components 1.4.1
3. matplotlib 3.7.1
4. pandas 1.5.3
5. plotly 5.13.1
6. PyQt5 5.15.9
7. numpy 1.24.2

**Структура программы**

Для выполнения программы используются различные виды файлов.

1. Датасеты “covid\_19\_data.csv” и “covid\_impact\_on\_airport\_traffic.csv”. Первый из них содержит в себе информацию о количестве заболеваемости коронавирусной инфекцией в конкретные даты (примерный период – начало 2020 – середина 2021) по местоположению. Он необходим для визуализации заболеваемости на интерактивной карте с помощью столбцов данных, представленных в данном датасете. Второй из них содержит в себе большое количество столбцов, с помощью которых мы можем получить графики линейной зависимости наполненности аэропортов от выбранного отрезка дат. Для данной задачи нам не требуется использовать все столбцы информации, так как фильтрация логична только при выборе страны, города, штата или названия аэропорта.

Файл для обмена данными “data.json” позволяет пользователю изменить URL локального сервера, не проникая в главный скрипт. В нём находится лишь URL локального сервера разработчика, что означает то, что не возникнет проблем с заменой на корректный URL.

Расположение файлов в папке проекта - /data

1. Вёрстка десктопного приложения “MainWindow.ui” облегчает работу, так как в QtDesigner есть возможность сверстать приложение и иметь представление о том, как будет расположены графические данные в графическом интерфейсе.

Расположение файла в папке проекта - /graphics

1. Скрипты “function.py” и “MainWindow.py”. Первый из них является шаблоном функции, которая создаёт графики зависимости заболеваемости от различных факторов. Второй их них требуется для разметки десктопного приложения. Этот файл является конвертированным “MainWindow.ui” с помощью команды “pyuic5 MainWindow.ui –o MainWindow.py”. Содержит в себе класс “MainWindow()”, который мы передаём в главный скрипт “main.py”. Как-либо изменять данный файл не рекомендуется.

Расположение файлов в папке проекта - /library

1. Скрипт “map.py” необходим для запуска сервера dash для выгрузки на него plotly.express интерактивной карты. Скрипт должен быть первоначально запущен с помощью anaconda prompt командой “python map.py” в директории, где находится данный скрипт. Сервер постоянно находится в активном режиме для того, чтобы бесперебойно показывать пользователю интерактивную карту. Если работа сервера будет прервана, то QWebEngineView не будет корректно показывать результат, так как он прямо транслирует содержимое сервера.

Расположение файла в папке проекта - /scripts/server

1. Скрипт “main.py” необходим для запуска главного окна приложения. В нем с помощью PyQt5 и всех скриптов описанных выше бесперебойно начинает работать пользовательский интерфейс. Из него мы получаем входные данные, который вводит пользователь в доступные поля ввода. Скрипт должен быть первоначально запущен с помощью anaconda prompt командой “python main.py” в директории, где находится данный скрипт.

Расположение файла в папке проекта - /scripts

Функция show\_plot() из скрипта functions.py создаёт и выводит на экран с помощью numpy фильтрации matplotlib графики по переданным в функцию значениям из пользовательского интерфейса. Если пользователь поставил галочку, то на графике появляется линия среднего значения. Также данная функция сохраняет графики в формате png-формате под индивидуальными названиями, в зависимости от выбранных критериев. Графики находятся в директории output.

Класс Ui\_MainWindow() из скрипта MainWindow.py передаёт в главную функцию вёрстку программы с помощью функции инициализирования setupUi() в программу передаются различные виджеты – QwebEngineView, QFrame, QLayout, QCheckbox, QCombobox, QButton. Функция retranslateUi() передаёт новые названия виджетам. Класс и функции в нём были сгенерированы автоматически.

Функция run\_server(), вызываемая в скрипте server.py из библиотеки dash позволяет запустить сервер в основном режиме, из чего и следует необходимость открывать две консоли для запуска десктопного приложения.

Все ниже описанные функции будут прописаны в скрипте main.py

Функция setup\_web\_engine() передаёт в MainWindow.py значение URL локального сервера. Так как номер порта на каждом компьютере отличается, можно произвести смену URL-адреса. Для этого данная функция помещена именно в main.py, вместо того, чтобы поместить адрес сразу в скрипт вёрстки приложения. Сам адрес пользователь может поменять, перейдя в json-файл в директории data.

Функция setup\_button() передаёт в functions.py, а конкретно в функцию show\_plot() значения, введённые в два поля с датами, выбор в выпадающем списке и наличие или отсутствие галочки, по нажатию кнопки.

Функция setup\_combo\_boxes() передаёт в функцию в functions.py значение первого ComboBox, чтобы во втором ComboBox выдавался список значений выбранного критерия. Список для критерия находится functions.py.

Функция handle\_combo\_box\_update() передаёт строку выбранного значения в функцию functions.py для фильтрации данных.

Функции get\_date\_from() и get\_date\_to() передаёт в функцию functions.py значения дат (тип данных – строка) в формате гггг-мм-дд для фильтрации по дате.

**Входные и выходные данные**

**Входные данные**

Входными данными являются следующие параметры, которые вводит пользователь через интерфейс:

1. Значения двух дат типа str
2. Значение критерия фильтрации типа str
3. Значение выбранного значения критерия фильтрации типа str
4. Значение выбора или невыбора вывода среднего значения типа bool

**Выходные данные**

Выходными данными являются десктопное приложение и графики зависимости при нажатии на кнопку.