**Министерство образования Республики Беларусь**

**Учреждение образования**

**«Брестский государственный технический университет»**

**Кафедра ИИТ**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОМУ ПРОЕКТУ ПО ДИСЦИПЛИНЕ**

**«Математические основы информационных систем»**

**Тема: «Реализация приложения сбора файлов системы организации пассажиропотока»**

**КП.ИИ-23.1-40-03-01**

**Листов:20**

**Выполнил:**

Студент 2-го курса,

ФЭИС,

Группы ИИ-23

Романюк А.П.

**Проверил:**

Козинский А.А.

**Брест 2024**

**АННОТАЦИЯ**

20 с., 4 рис., 1 табл., 8 ист. лит.

Целью данного курсового проекта является создание приложения для сбора файлов и обработки информации о пассажиропотоке с использованием облачного сервиса Google Disk.

Курсовой проект включает в себя такие разделы как анализ предметной области по хранению данных, отбор средств обработки данных, разработка структуры приложения и реализация компонентов, тестирование.

**Учреждение образования**

**Брестский государственный технический университет**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Факультет | | ЭИС | | | | Кафедра | | ИИТ |
|  | |  | | | |  | |  |
| **УТВЕРЖДАЮ** | | | | | | | | |
| Зав. кафедрой | | |  | | |  | | |
|  | | | (подпись) | | |  | | |
| «» |  | | | | 2024 г. | | | |
|  |  | | | |  | | | |
| **З А Д А Н И Е** | | | | | | | | |
| **по курсовому проектированию** | | | | | | | | |
|  | | | | | | | | |
| Студенту | | Романюку Алексею Петровичу (гр. ИИ-23) | | | | | | |
| 1. Тема проекта | | | | Реализация приложения сбора файлов системы прогнозирования | | | | |
|  | | | | пассажиропотока | | | | |
| 2. Сроки сдачи студентом законченного проекта | | | | | | | 15.05.2024 г. | |
| 3. Исходные данные к проекту:  Объект исследования: Данные о пассажиропотоке хранящиеся на облаке Google Disk  Цель исследования: реализовать приложение сбора файлов системы прогнозирования пассажиропотоков.  Особенности разработки:  - язык программирования – Python  - среда разработки – PyCharm  - библиотеки Python – google-api-python-client, BeautifulSoup4, Pandas, Datetime  Система должна обеспечивать:  -запуск приложения и доступ к файлам конфигурации  -проверку доступа к глобальной сети интернет  -Загрузку данных хранящихся в виде файлов необходимого формата в облаке Google  -Обработку полученных данных  -Загрузку итоговых данных в облако Google | | | | | | | | |
| 4. Содержание расчетно-пояснительной записки (перечень разрабатываемых вопросов) | | | | | | | | |
| Введение | | | | | | | | |
| 1.Анализ предметной области по хранению данных | | | | | | | | |
| 2. Отбор средств обработки данных | | | | | | | | |
| 3. Разработка структуры приложения и реализация компонентов | | | | | | | | |
| 4. Тестирование | | | | | | | | |
| 5. Заключение | | | | | | | | |
| 6. Список использованных источников | | | | | | | | |
|  | | | | | | | | |
|  | | | | | | | | |
|  | | | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 6. Консультанты по проекту (с указанием относящихся к ним разделов проекта) | | | | | | |
|  | | | | | | |
|  | | | | | | |
| 7. Дата выдачи задания | | | 12.02.2024 г. | | | |
| 8. Календарный график работы над проектом на весь период проектирования (с указанием | | | | | | |
| сроков выполнения и трудоемкости отдельных этапов) | | | | |  | |
| Раздел 1: до 20.02.2024: 5% | | | | | | |
| Раздел 2: 20.02.2024 – 05.03.2024: 15% | | | | | | |
| Раздел 3: 05.03.2024 – 08.04.2024: 50% | | | | | | |
| Раздел 4: 08.04.2024 – 16.04.2024: 10% | | | | | | |
| Оформление проекта: 16.04.2024 – 26.04.2024: 20% | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | **Руководитель** | |  | |  |
|  | |  | | (подпись) | |  |
|  | | | | | | |
| Задание принял к исполнению (дата) | | | |  | |  |
|  | | | | | | |
| (подпись студента) |  | | | | |  |

**СОДЕРЖАНИЕ**

[Введение 6](#_Toc167051117)

[1. Анализ предметной области по хранению данных 8](#_Toc167051118)

[2. Постановка задачи и отбор средств реализации «приложения сбора файлов системы организации пассажиропотока» 12](#_Toc167051119)

[3. Разработка структуры приложения 15](#_Toc167051120)

[4. Тестирование 16](#_Toc167051121)

[5. Заключение 18](#_Toc167051122)

[Список литературы 20](#_Toc167051123)

# Введение

В наше время все больше и больше людей начинает пользоваться общественным транспортом и эффективное управление перевозками становится все более критическим для городских организаций и предприятий. В контексте города Бреста, о проблеме не эффективного графика движения транспорта, и неудобстве перемещения по городу в часы пик на общественном транспорте сообщают газеты СБ Беларусь [1] и БрестСити [2].

Коммунальное транспортное унитарное предприятие «Брестгортранс» [5] играет ключевую роль в организации и контроле пассажирских перевозок. Несмотря на уже проводимые мероприятия по контролю пассажиропотока, существующие подходы требуют дополнительной расширенной функциональности и оптимизации для повышения эффективности.

Исходя из необходимости оптимизации движения и распределения транспорта, а также анализа нынешней ситуации, нами определена следующая тема для исследования «Реализация приложения сбора файлов системы организации пассажиропотока». Приложение будет являться частью системы по обработке данных о состоянии пассажиропотоков.

Объектом исследования являются данные о пассажиропотоке в городе Бресте.

Предметом исследования выбраны возможные способы обработки информации о пассажиропотоке при помощи Google API.

**Цель** данного исследования: создать приложение для обработки информации о пассажиропотоке, хранящейся в виде файлов в облачном хранилище Google Drive. Приложение должно выполнять такие функции как: скачивание файлов из облака, сборка и дополнение информацией данных в файлах, валидация данных, объединение всех файлов и загрузка на облачное хранилище.

Для реализации данной цели необходимо решить следующие задачи:

1. Выполнить анализ способов обработки данных облачного хранилища Google

2. Осуществить постановку задачи на создание приложения для сбора файлов системы прогнозирования и отбор средств ее реализации

3. Разработать архитектуру локального приложения и реализовать его компоненты

4. Выполнить тестирование приложения

# 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ ПО ХРАНЕНИЮ ДАННЫХ

Основные методы хранения информации охватывают ряд технологий [3], каждая из которых имеет свои особенности и ограничения. Рассмотрим каждый из них более детально:

1. Жесткие диски: Этот классический метод хранения данных предлагает надежность и относительно низкую стоимость за гигабайт, что делает его привлекательным выбором для больших объемов информации. Однако, ограничения на скорость доступа и необходимость физического присутствия устройства с диском могут затруднить обмен данными между пользователями.

2. Flash-накопители: Скорость чтения и записи на флэш-накопителях значительно выше, чем на обычных жестких дисках, что делает их предпочтительным выбором для операций, требующих быстрого доступа к данным. Однако, их емкость часто ограничена, и, как и в случае с жесткими дисками, требуется физическое присутствие для передачи информации.

3. Оптические диски: Хотя оптические диски имеют свои преимущества в виде долговечности и относительно низкой стоимости носителей, их емкость существенно ограничена по сравнению с более современными технологиями. Кроме того, не все современные устройства оборудованы оптическими приводами, что ограничивает их применимость.

4. Облачные хранилища: В последние годы облачные технологии стали все более популярными благодаря своей универсальности и гибкости. Google Disk, в частности, предлагает простой доступ к данным из любой точки мира при наличии интернета, а также предоставляет разработчикам API для взаимодействия с файлами через программный код. Однако, для использования облачных хранилищ требуется надежное интернет-соединение, что может оказаться недоступным в некоторых ситуациях.

5. Гибридные решения: Некоторые предприятия предпочитают гибридные подходы, комбинируя облачные и локальные хранилища. Это позволяет им совмещать преимущества обоих методов, обеспечивая гибкость, безопасность и высокую доступность данных. Однако, такой подход требует дополнительных усилий по управлению инфраструктурой и обеспечению совместимости между различными системами.

Также стоит учитывать аспекты из [4]:

1. Безопасность данных: Важным аспектом хранения данных является их безопасность. При выборе метода хранения необходимо учитывать меры защиты информации от несанкционированного доступа, утечек данных и других угроз. Облачные хранилища, например, часто предлагают широкий спектр механизмов шифрования и аутентификации, что делает их привлекательным выбором для организаций, ценящих безопасность.

2. Скорость доступа и производительность: В зависимости от конкретных потребностей приложения, важно учитывать скорость доступа к данным и их обработку. Например, для приложений, требующих высокой скорости чтения и записи, могут быть предпочтительны быстрые флэш-накопители или специализированные облачные решения с высокой производительностью.

3. Стоимость и экономическая эффективность: Помимо технических аспектов, важно также оценить экономическую целесообразность выбранного метода хранения данных. Это включает в себя как непосредственные затраты на приобретение и поддержку инфраструктуры, так и прогнозируемые операционные расходы на обслуживание и масштабирование системы.

При рассмотрении различных методов хранения данных для курсового проекта было принято решение остановиться на облачном хранилище, а именно на Google Disk. Это решение обосновывается несколькими ключевыми факторами [9].

Во-первых, облачные хранилища, такие как Google Disk, обеспечивают универсальный доступ к данным из любой точки мира при наличии интернета. Это обеспечивает удобство использования для всех пользователей приложения, независимо от их местоположения или используемого устройства.

Во-вторых, облачные хранилища предлагают гибкость и возможность масштабирования по мере роста потребностей. Мы можем легко увеличить объем хранимых данных или добавить новые функции без необходимости приобретения и настройки нового оборудования. Это позволяет нам адаптироваться к изменяющимся требованиям и масштабировать приложение в соответствии с ростом пользовательской базы.

Третий фактор - безопасность и надежность. Ведущие облачные провайдеры, включая Google, обеспечивают высокий уровень защиты данных, включая механизмы шифрования и многоуровневые системы аутентификации. Это гарантирует надежную защиту конфиденциальности и целостности информации нашего приложения.

Наконец, удобство разработки и интеграции также является значимым фактором. Google предоставляет разработчикам API для взаимодействия с файлами на Google Disk, что упрощает процесс интеграции облачного хранилища в наше приложение. Мы можем использовать его функциональность, такую как загрузка, скачивание и синхронизация файлов, при помощи программного кода, что позволяет нам сосредоточиться на создании качественного пользовательского опыта.

Таким образом, выбор облачного хранилища обусловлен его универсальным доступом, гибкостью, безопасностью и удобством интеграции, что позволит нам создать эффективное и удобное приложение для пользователей.

Для выбора облачного хранилища рассмотрим сравнение бесплатных вариантов, таких как Яндекс диск и Google Disk (см. [6]) . Результаты сравнения приведены в таблице 1.1.

Таблица 1.1. Сравнение облачных сервисов

| **GOOGLE ДИСК** | **ЯНДЕКС ДИСК** |
| --- | --- |
| Он предлагает 15 ГБ свободного места для хранения. | Он предлагает 10 ГБ свободного места для хранения. |
| Он предоставляет ограниченное максимальное пространство для платного хранения. | Он также предоставляет ограниченное максимальное пространство для платного хранения. |
| API разработчика - OAuth2. | API разработчика - это REST, WebDAV и SDK. |
| У него нет ограничений по трафику или ширине полосы пропускания. | Ограничение по трафику или ширине полосы составляет 200% от объема хранилища в день. |
| Максимальный размер файла здесь составляет 5 ГБ. | Здесь максимальный размер файла составляет 2 ГБ при загрузке через Интернет и 50 ГБ при загрузке через приложение. |
| У него больше пользователей. | У него сравнительно меньшее количество пользователей. |

Использование формата .csv (comma-separated values) для хранения данных в нашей системе обладает рядом существенных преимуществ, которые делают его идеальным выбором для нашего проекта. Основные достоинства .csv файлов заключаются в их простоте, универсальности и легкости интеграции с различными системами и инструментами. .csv файлы имеют простой и интуитивно понятный формат. Данные записываются в виде строк, где каждая строка представляет одну запись, а значения в строках разделяются запятыми. Это делает .csv файлы легко читаемыми как для людей, так и для машин.

Формат .csv поддерживается практически всеми табличными и базами данных, включая такие популярные инструменты как Microsoft Excel, Google Sheets, и большинство систем управления базами данных (СУБД). Это позволяет без труда импортировать и экспортировать данные между различными приложениями и платформами. В отличие от более сложных форматов данных, .csv файлы не содержат метаданных или структурных элементов, таких как теги или описатели типов данных. Это позволяет сохранять большие объемы данных в компактном виде, экономя пространство на диске и уменьшая время передачи данных. Работа с .csv файлами не требует специализированного программного обеспечения. Большинство языков программирования, включая Python, R, и Java, имеют встроенные библиотеки для чтения и записи .csv файлов, что облегчает их обработку и анализ. Исходя из вышеизложенного будем использовать .csv формат файла.

Так как в нашей системе выбрано хранить данные в .csv формате и взаимодействовать с ними остановим свой выбор на Google disk. Приложению же надо будет собирать файлы с диска

Постановка задачи на создание «приложения сбора файлов системы организации пассажиропотока» выполнена в следующей главе.

Выводы по главе 1:

Результатом анализа способов хранения данных являются следующие положения.

В качестве средства хранения информации нами выбрано облачное хранилище Google Disk.

Записи данных о пассажиропотоках будут храниться в формате .csv.

# 2. ПОСТАНОВКА ЗАДАЧИ И ОТБОР СРЕДСТВ РЕАЛИЗАЦИИ «ПРИЛОЖЕНИЯ СБОРА ФАЙЛОВ СИСТЕМЫ ОРГАНИЗАЦИИ ПАССАЖИРОПОТОКА»

Согласно выводам, полученным в главе 1. Необходимо создать приложение для обработки данных хранящихся в облачном хранилище Google Disk в формате .csv. В соответствии с этим представим следующую постановку задачи.

1. Общие сведения о приложении

Наименование: «Приложение сбора файлов системы организации пассажиропотока».

Основание для разработки: Необходимость оптимизации процессов управления пассажиропотоками. Задание на курсовую работу.

2. Цели и назначение создания приложения

Цель создания: Автоматизация процесса обработки информации, полученной при фиксации данных о пассажиропотоках на остановках г.Бреста и хранящихся в облачном хранилище в виде файлов формата csv.

Назначение: Приложение предназначено для сбора данных о пассажиропотоках. Приложение должно обеспечивать доступ к данным и их сбор на локальном компьютере для дальнейшей обработки.

3. Характеристика объектов автоматизации

Объектом автоматизации является процесс обработки данных о пассажиропотоках, включающий:

– доступ к облачным ресурсам;

– сбор данных о пассажиропотоках из облака;

– обработку и анализ собранных данных;

– хранение и предоставление результатов анализа.

4. Требования к приложению

Функциональные требования:

– запуск приложения и доступ к файлам конфигурации.

– проверка доступа к глобальной сети Интернет.

– загрузка данных из облачного хранилища Google.

– обработка полученных данных.

– загрузка итоговых данных обратно в облако Google.

Нефункциональные требования:

– надежность и устойчивость работы приложения обеспечивается средствами операционной системы.

– защита данных и безопасность доступа обеспечивается средствами Google Disk.

– требований к пользовательскому интерфейсу не предъявляется.

– приложение должно обеспечивать возможность обработки большого объема данных на различных ресурсах Google Disk.

5. Состав и содержание работ по созданию приложения

– разработка технического задания.

– проектирование архитектуры.

– разработка и тестирование программного обеспечения.

– внедрение и настройка приложения.

– обучение пользователей и техническая поддержка.

6. Порядок разработки приложения

1. Стадия подготовки (разработка ТЗ).

2. Стадия проектирования (создание архитектуры и дизайна системы).

3. Стадия разработки (программирование, тестирование модулей).

4. Стадия внедрения (установка системы, настройка, интеграция).

5. Стадия эксплуатации (поддержка и сопровождение системы).

7. Порядок контроля и приемки приложения

– первичное тестирование системы разработчиком.

– исправление выявленных ошибок и недочетов.

– финальное тестирование и приемка приложения в эксплуатацию.

8. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу автоматизированной системы в действие

– обеспечение доступа к глобальной сети Интернет.

– настройка учетных записей и прав доступа.

– подготовка файлов инициализации

– обучение работе с системой.

9. Требования к документированию

– руководство пользователя.

– руководство администратора системы.

– техническая документация (описание архитектуры системы, исходный код, инструкции по установке и настройке).

Для разработки данного приложения выбран язык программирования Python по ряду причин. Как указано в [8] во-первых, Python известен своей простотой в использовании и читаемостью кода, что упрощает разработку и поддержку приложений. Во-вторых, наличие богатого экосистема библиотек и фреймворков, включая библиотеку Google API для Python, облегчает интеграцию с различными сервисами и API, такими как Google Disk. Кроме того, Python активно поддерживается сообществом разработчиков, что обеспечивает доступ к обширным ресурсам и документации для решения разнообразных задач. Наконец, Python является платформонезависимым языком программирования, что позволяет запускать приложение на различных операционных системах без необходимости изменения исходного кода, что важно для обеспечения удобства использования и распространения приложения.

Выводы по главе 2:

В качестве средства реализации приложения выбраны:

язык Python для собственно приложения;

Google Disk для хранения данных;

Google API для взаимодействия приложения с Google Disk.

# 3 РАЗРАБОТКА СТРУКТУРЫ ПРИЛОЖЕНИЯ

В рамках курсового проекта была разработана структура взаимодействия приложения с Google Drive, посредством Google API.

Общая структура приложения показана на рисунке 3.1

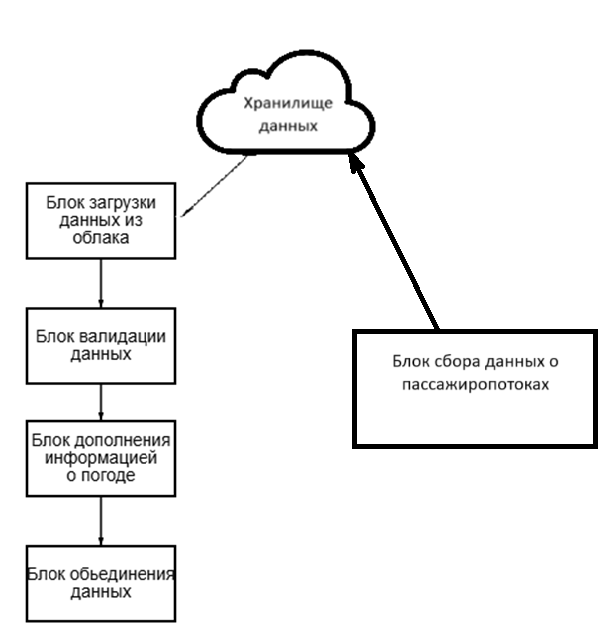


Рисунок 3.1 - Общая структура приложения

Основными компонентами приложения являются:

GoogleApi.py: Модуль отвечающий за взаимодействие с Google API для работы с файлами и папками в Google Drive.

Main.py: Основной файл, в котором содержатся основные функции обработки скачанных данных.

Run.but: Файл автоматической подкачки необходимых библиотек и запуска приложения для операционной системы Windows.

Run.sh: Файл автоматической подкачки необходимых библиотек и запуска приложения для операционной системы Linux.

Также для работы приложения требуется файл конфигурации под названием folders\_id.csv содержащий адреса папок с данными.

Остальное взаимодействие с файловой системой приложение должно осуществлять без участия пользователя.

Выводы по главе 3:

Структура приложения включает модули:

GoogleApi.py реализация взаимодействия с гугл диском.

Main.py основной код приложения.

Run.but файл для запуска приложения на Windows.

Run.sh файл для запуска приложения на Linux.

# 4. ТЕСТИРОВАНИЕ

В рамках курсового проекта по разработке приложения используется run.bat файл для запуска, это освобождает от необходимости пользовательского интерфейса. Содержание run.bat файла представлено во фрагменте 4.1:

Фрагмент 4.1. Содержание файла run.bat

@echo off

REM Запуск другого установка библиотек

pip install -r requirements.txt > nul

REM Запуск Python скрипта

python main.py

Проект тестировался на компьютере с установленным интерпретатором языка программирования Python.

Перед началом тестирования необходимо настроить приложение через файл folders\_id.csv. Необходимо вписать в файл id папок Google Disk из которых будут браться данные(см. фрагмент 4.2).

Фрагмент 4.2. Содержание файла folders\_id.csv

id

1OEon\_6lH94B6TupvVeicEwf8cc1vEIfL

1OEon\_6ldfbdfbTupvVeicEwf8cc1vEIfL

234Eon\_6lergeerhherherhEwf8cc1vEIfL

Далее запускаем run.bat и ожидаем пока программа подключит библиотеки и загрузит данные.

Приложение загружает данные в папку под названием download, внутри папки находятся еще две папки, где в папку img сохраняются изображения, а в папку data данные из о пассажиропотоках из Google Disk.

Обработанные данные сохраняются в папку upload в виде файла Data.csv.

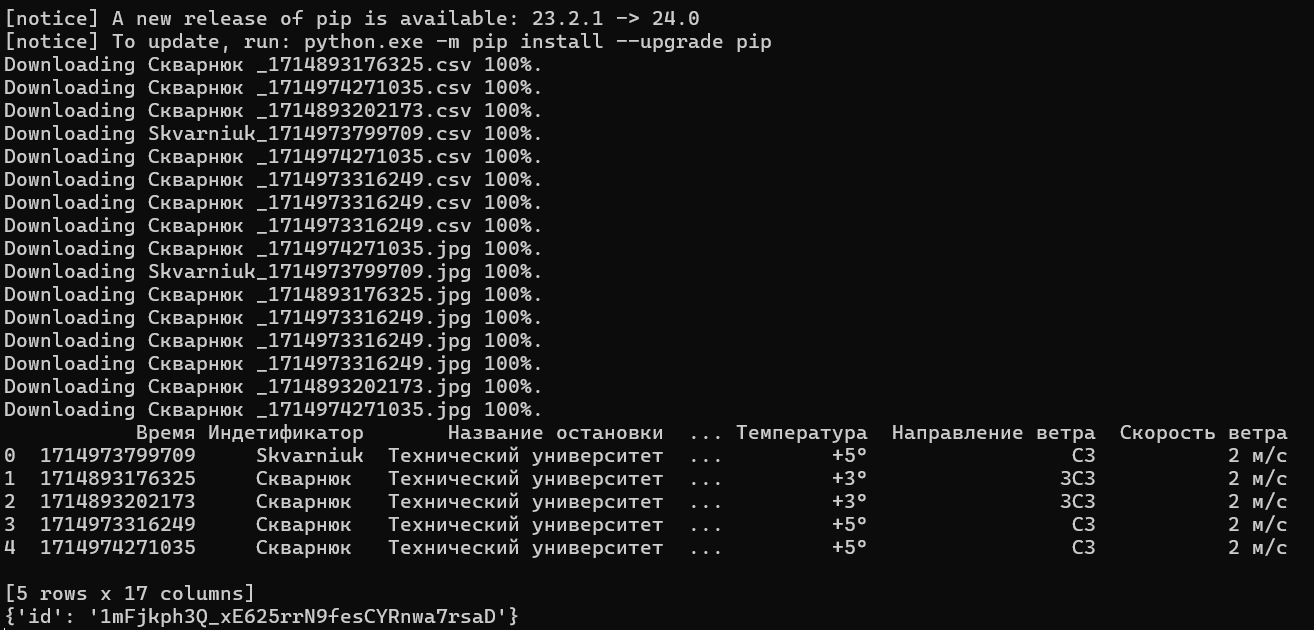


Рисунок 4.1. Процесс работы приложения

Приложение загрузило все файлы и теперь с ними можно работать с компьютера, или открыть на Google Disk

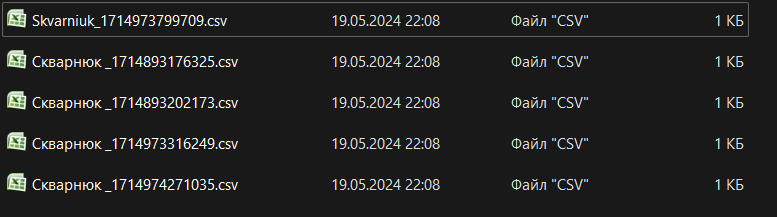
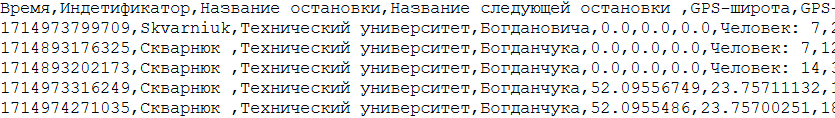


Рисунок 4.2 Файлы загруженные на компьютер

Фрагмент 4.3. Содержание итогового файла data.csv



Выводы по главе 4:

Приложение протестировано на компьютерах с различными операционными системами (Windows 11, Fedora 38).

В условиях тестирования приложение обеспечивает сбор информации csv файлов облачных хранилищ пользователей.

# 5 ЗАКЛЮЧЕНИЕ

В ходе выполнения курсового проекта разработано приложение сбора файлов системы прогнозирования пассажиропотоков.

В результате анализа способов хранения данных в качестве средства хранения информации выбрано облачное хранилище Google Disk. Данные о пассажиропотоках хранятся в формате .csv. Данные положения явились результатом анализа представленным в главе 1.

Средствами реализации выбраны Python для собственно приложения, Google Disk для хранения данных, Google API для взаимодействия приложения с Google Disk. Подробный отбор средств реализации описан в главе 2.

Структура приложения представлена модулями GoogleApi.py Main.py Run.but. Назначение модулей описано в главе 3.

Работоспособность приложения проверена на компьютерах с операционными системами Windows 11, Fedora 38. Тестирование показало, что приложение обеспечивает сбор информации csv файлов облачных хранилищ пользователей.

Тем самым все поставленные задачи курсового проектирования решены в полном объеме. Цель состоящая в реализации приложения сбора файлов системы прогнозирования пассажиропотоков достигнута.

В будущем приложение может быть дополнено модулями для валидации данных о пассажиропотоках и их прогнозировании с использованием нейронных сетей.

# СПИСОК ЛИТЕРАТУРЫ

1. Виктор ПОНОМАРЕВ Новый проект «Городской транзит»: оцениваем состояние общественного транспорта в областных центрах. Брест / Виктор ПОНОМАРЕВ [Электронный ресурс] // www.sb.by : [сайт]. — URL: https://www.sb.by/articles/brest-komu-taksi-na-dubrovku.html?ysclid=luqkkwyvu2152492165 (дата обращения: 23.04.2024).

2. Виктор ПОНОМАРЕВ Есть претензии... Жителей Бреста спросили о работе маршруток и общ.транспорта / Виктор ПОНОМАРЕВ [Электронный ресурс] // brestcity.com : [сайт]. — URL: https://brestcity.com/blog/est-pretenzii (дата обращения: 05.05.2024).

3. Куликов А.А. АНАЛИЗ СИСТЕМ ХРАНЕНИЯ ДАННЫХ / Куликов А.А. [Электронный ресурс] // cyberleninka.ru : [сайт]. — URL: https://cyberleninka.ru/article/n/analiz-sistem-hraneniya-dannyh-1?ysclid=lvtzigohu4978348041 (дата обращения: 05.05.2024).

4. Шевцов Вадим Юрьевич, Абрамов Евгений Сергеевич Анализ современных систем хранения данных / Шевцов Вадим Юрьевич, Абрамов Евгений Сергеевич [Электронный ресурс] // cyberleninka.ru : [сайт]. — URL: https://cyberleninka.ru/article/n/analiz-sovremennyh-sistem-hraneniya-dannyh?ysclid=lvtyj4nkyp354706448 (дата обращения: 01.05.2024).

5. Официальный сайт Брестгортранс / [Электронный ресурс] // brestgortrans.by : [сайт]. — URL: https://brestgortrans.by/ (дата обращения: 05.05.2024).

6. pp\_pankaj Difference between Google Drive and Yandex Disk / pp\_pankaj [Электронный ресурс] // geeksforgeeks.org : [сайт]. — URL: https://www.geeksforgeeks.org/difference-between-google-drive-and-yandex-disk/ (дата обращения: 05.05.2024).

7. Google Drive API overview / [Электронный ресурс] // google.com : [сайт]. — URL: https://developers.google.com/drive/api/guides/about-sdk?hl=en (дата обращения: 05.05.2024).

8. Марк Лутц Изучаем Python [Текст] / Марк Лутц — 5-е изд. : Вильямс, 2019 — 864 c.

9. Возможности Google Drive / [Электронный ресурс] // Google.by : [сайт]. — URL: https://www.google.by/intl/ru/drive/#features (дата обращения: 20.05.2024).