Руководство для пользования классом KATO File Sorter для ЭПХУ

Департамент контроля качества данных и развития коммуникаций

Бюро национальной статистики АСПИР РК

Черкасов Б. Ю.

2025

1. Описание

Приложение KATO File Sorter предназначено для автоматической сортировки, обработки и объединения данных из Excel-файлов в структуре, где документы распределены по папкам согласно приложениям (1-қосымша, 2-қосымша и т.д.).

2. Основные функции

- Рекурсивный обход всех вложенных папок.
- Удаление ранее сгенерированных папок като_файлы, чтобы избежать дублирования.
- Обработка Excel-файлов и извлечение данных по шаблону.
- Формирование итоговой директории с Excel-таблицами с форматированием по каждому KATO, полученных из приложений.

3. Описание входной структуры

Пример структуры входных данных:

```
project_directory/
 main.py (скрипт с классом, либо можно отдельно класс описать)
 kato_sorter.py (опционально, код класса в отдельном модуле)
 РЕГИОН1/
   1-қосымша___.xlsx
   2-қосымша___.xlsx
 РЕГИОН2/
    1-қосымша.xlsx
    ...
```

4. Установка зависимостей

Для корректной работы необходимо установить зависимости. Создайте файл requirements.txt со следующим содержимым:

```
pandas==2.3.0
openpyxl==3.1.5
```

Установите их командой:

```
pip install -r requirements.txt
```

5. Алгоритм работы

1. Установить интерпретатор Python версии 3.10 или выше.
2. Установить удобную среду разработки, например:
   - PyCharm Community / Professional Edition (от JetBrains),
   - Visual Studio Code (VS Code).
3. Создать папку проекта, внутри которой:
   - Либо разместить весь код в одном файле main.py,
   - Либо разделить код на модуль kato_sorter.py и файл запуска main.py, где будет вызов класса KATOFileSorter.
4. Установить зависимости из файла requirements.txt:
   ```
   pip install -r requirements.txt
   ```
5. Перенести все папки регионов с данными по приложениям (например, 1-қосымша, 2-қосымша, ...) в корневую директорию проекта, рядом со скриптом.
6. Убедиться, что в папках нет лишних вложенных директорий или старых экспортов (като_файлы и др.).
7. Запустить скрипт:
   ```
   python main.py
   ```
8. Дождаться завершения работы (ориентировочно 10–15 минут), после чего появится структура вида:

```
итоговые_файлы/
 Абай/
   151011.xlsx
   ...
 Жетісу/
   121007.xlsx
   ...
```

6. Запуск

1. Структура файла-запуска

Пример содержимого файла main.py:

```
from kato_sorter import KATOFileSorter

if __name__ == '__main__':
    sorter = KATOFileSorter(input_dir='.')
    sorter.delete_kato_subfolders()
    sorter.process_files()
    sorter.save_kato_files()
```

В процессе выполнения:

- Все подпапки като_файлы (случайные, ненужные), если они есть, будут автоматически удалены.
- Скрипт обойдёт все найденные Excel-файлы и распределит записи по КАТО-кодам.

2. Запуск программы

Запустите основной скрипт (например, main.py):

```
python main.py
```

После завершения будет создана директория итоговые.

7. Структура выходных файлов

Итоговые файлы содержат:

- Отформатированные столбцы
- Удалённые дубликаты (за исключением случаев с 1-қосымша)
- Объединённые данные по каждому приложению
- Все записи рассортированы по КАТО каждого акимата, подразделения (села), добавляются на отдельных Excel-листах, именованных под «Приложение-НОМЕР» по возрастанию.

8. Замечания

- Повторная обработка без удаления подпапок или ненужных файлов может привести к дублированию данных.
- Для чистого запуска рекомендуется удалять старые като_файлы внутри папок.