

Авторы

Имя	Должность	Контакт
Лаврова Елена Владимировна	проф., д.т.н., директор Учебно-научного института информационных технологии	UNIT-PSTU@yandex.ru
Иванов Григорий Александрович	Ассистент кафедры Информационной безопасности	ivanov.g.a@ro.ru

§ 1. Вводный контекст и обоснование

В условиях современной образовательной среды, характеризующейся динамичным пересмотром учебных программ, гибким распределением ресурсов и растущими требованиями к персонализации образовательного процесса, необходима **синергетическая** система, способная **интегрировать** и **анализировать** сложные структурные объекты — таблицы Excel с многослойными заголовками, объединенными ячейками и «параллельными» слоями информации. Предложенный проект реализует **мультимодальный** конвейер, в котором **модуль парсинга** (на основе *openpyxl* и *pandas*) извлекает **структурированные данные** из *XLSX*, корректно обрабатывая **объединенные ячейки** и **иерархические заголовки**, после чего преобразует их в **иерархический JSON** (стратегия *schema-first*), отражающий семантику «лист → день → неделя → пара → группы/предметы».

В дополнение к парсеру реализован **асинхронный Telegram-бот** (на базе *aiogram*), который выступает в роли **интерактивного посредника** (шаблон «Посредник») между пользователем и **сервисной логикой**. Бот поддерживает **командный интерфейс** /start, /test, /set, /help, а также **пользовательскую обработку** загрузки *XLSX* от администраторов, **публикацию** персонального расписания по ключу (*search_key*), **полное отображение** расписания и **уведомление о перерывах** через *ScheduleNotifier*.

§ 2. Технологическая архитектура

• 2.1. Парсинг и трансформация

- **logger.py** — «обёртка» над стандартным *logging* с оператором << для отправки сообщений, включающая **систему дисциплин** (*DisciplineLogger*) для фиксации ключей шаблона, незаполненных меток и успешных замен.
- **parse_xlsx.py** — интерфейс командной строки, использующий *argparse* для выбора файла/директории, пути сохранения *JSON* и вызова *ExcelHeaderParserXLSX*. Внутри реализован **объектный подход** с кэшированием диапазонов **объединенных ячеек**, **рекурсивным сканированием** заголовков, **сглаживанием** и **flatten_headers** для формирования плоского *DataFrame*, а затем **иерархического JSON** (см. *extract_json_hierarchical*).

- **test_pyx1.py** / **КнигаXLSX** – альтернативный, «русскоязычный» парсер, демонстрирующий **разностный подход** к обработке merged-cells, **генерацию** JSON-структуры и её сериализацию в файл.

2.2. Телеграм-бот и бизнес-логика

- Модуль **bot.py** (или единый файл) содержит **компонентный стек**:
- **UserInterface** — уровень пользовательского интерфейса, реализующий асинхронные методы `handle_command`, `show_message`.
- **BusinessLogic** — бизнес-уровень, обрабатывающий команды `/set ...`, проверяющий **активность** пользователей, формирующий запросы (`schedule_query`, `all_schedule_query`, `help_query`).
- **DataAccess** — уровень доступа к данным (SQLite, JSON-файл расписания), реализующий **кэширование** `schedule`, **фильтрацию** по `search_key`, **поиск** по группам/преподавателям.
- **ScheduleNotifier** — сервис рассылки уведомлений о перерывах, использующий **асинхронный таймер** `asyncio.sleep(60)` и **проверку** `is_activated`, `autosend`.
- **ConcreteMediator** — реализация шаблона посредника, связывающая все уровни и управляющая событиями (`user_command`, `test_notification`, `update_settings`, `activate_user`).

2.3. Хранение и настройка

- `raspysnye.json` — **текущий кэш расписания**, обновляемый после каждого парсинга.
- `usersettings.db` — база данных пользователей **SQLite**, автоматически создаваемая при инициализации `UserSettingsDB`.
- `secret.txt` — **однострочный токен** бота, хранится вне репозитория и считывается при запуске.
- Логи (`parse_xlsx.log`, `xlsx_processing.log`, `test_pyx1.log`, `discipline.log`) – **асинхронные** записи, включающие **контекст** вызова (файл, строка).

§ 3. Функциональные возможности

Возможность	Техническое реализованное решение	Техническая оценка
Надёжный разбор заголовков	Рекурсивный анализ merged-cells, проверка <code>is_merged_cell</code> , <code>merged_value</code> .	Высокая точность, но потенциальный <code>RecursionError</code> при глубокой вложенности.
Конвертация в DataFrame	<code>openpyxl + pandas</code> , <code>merged_value</code> → <code>DataFrame</code> .	Эффективно для средних файлов (<10k строк).

Возможность	Техническое реализованное решение	Техническая оценка
Иерархический JSON	<code>extract_json_hierarchical</code> → вложенный словарь → <code>json.dump</code> .	Гибкая структура, но требует корректного ключа <code>search_key</code> .
Telegram-бот	<code>aiogram</code> + Mediator; команды <code>/set</code> , <code>/help</code> , <code>/test</code> , расписание, всё расписание.	Асинхронно и ориентировано на пользователя.
Рассылки о перерывах	<code>ScheduleNotifier._send_break_notifications</code> + асинхронный запуск. Блокировка.	Периодический тикер (60 с), но без тайм-зоны.
Логирование	<code>Logger</code> + <code>DisciplineLogger</code> + <code>logger.py</code> .	Псевдосинхронно ; потребует ротацию.

§ 4. Инструкция по развёртыванию и эксплуатации

1. Инициализация окружения

```
python3 -m venv .venv
source .venv/bin/activate Windows: .venv\Scripts\activate
pip install -r requirements.txt
```

- 1. Конфигурация токена** — создайте в корневом каталоге файл `secret.txt` и поместите в него токен.

2. Парсинг

```
python parse_xlsx.py -f example.xlsx -j ./out_json
python parse_xlsx.py -d ./xlsx_files -j ./out_json
```

Для массовой обработки создается резервная копия JSON-файлов (`backup_ГГГГммдд_ЧММСС.zip`). 4. **Запуск бота** `bash` питон `bot.py`

Бот доступен по ``/start``, ``/test``, ``/set``, ``/help``, а также реагирует на ключевые слова в сообщениях.

Замечание

Приведённый проект представляет собой комплексный интеграционный модуль, объединяющий обработку данных, веб-интерфейс и интерактивную коммуникацию. Его архитектура, основанная на паттернах Mediator, Repository, Service, обеспечивает модульность, масштабируемость и прозрачность в управлении расписаниями.