

# Разработка парсера для автоматизации работы с учебными планами факультета компьютерных наук Воронежского государственного университета

М. Н. Быков, email: m.byckoff2015@yandex.ru<sup>1</sup>

Е. А. Копытина, email: zhemkaterina@yandex.ru<sup>1</sup>

<sup>1</sup> Воронежский государственный университет

**Аннотация.** В данной статье предложен подход для разработки парсера учебных планов факультета компьютерных наук Воронежского государственного университета, требуемого для создания базы данных, которая ляжет в основу автоматизации формирования части документа «Основная профессиональная образовательная программа».

**Ключевые слова:** электронные документы, база данных, учебные планы, автоматизация рутинных задач.

## Введение

За последние десятилетие мы всё более явно ощущаем наступление в истории человечества новой эры – Информационной. Объём данных, с которым мы встречаемся каждый день и который необходим нам для выполнения наших профессиональных обязанностей, неуклонно растёт.

Разумеется, для эффективного взаимодействия с подобным объёмом информации, требуются соответствующие инструменты по их обработке. на помощь приходят признанные средства, такие как Microsoft Excel или LibreOffice Calc. Однако, зачастую перед нами ставятся нетривиальные или комплексные задачи, выполнение которых при помощи готовых решений становится неэффективно, ведь большую часть работы приходится делать вручную, как показано в [1-8].

Именно в таких ситуациях максимальную пользу приносят инструменты, сконструированные специально для выполнения поставленной задачи. Одним из примеров использования таких инструментов является составление учебных планов для факультета компьютерных наук Воронежского государственного университета. Для каждого направления и года обучения формируется книга Excel, содержащая страницы «Титул», «График», «План», «Компетенции», и так далее.

Для эффективной организации учебного процесса, необходимо организовать автоматический перенос данных из книги Excel в базу данных. Конечно, есть вариант выполнять данные операции вручную,

однако таким образом придется потратить слишком много времени на рутинную работу. Именно для такой ситуации пригодится описываемый в рамках данной статьи парсер данных, время работы которого значительно выше. Этот проект нацелен на ускоренное формирование полноценной базы данных, исходя из информации, находящейся в таблицах книги Excel, что делает его крайне перспективным инструментом в рамках организации работы с информацией.

### **1. Постановка задачи**

К описываемому в статье проекту по реализации скрипта-парсера для создания и заполнения реляционной базы данных, предназначенной для хранения учебных планов факультета компьютерных наук Воронежского государственного университета, были выдвинуты следующие требования:

- Чтение информации из таблиц формата «xlsx»;
- Возможность подключения к базе данных напрямую для дальнейшей отправки сгенерированных SQL-запросов;
- Автоматическое преобразование информации из таблиц к легкочитаемому виду.

### **2. Средства реализации**

Проект реализуется посредством создания базы данных для хранения и загрузки в неё отношений, скрипты создания которых написаны на диалекте PostgreSQL.

В процессе работы были использованы следующие технологии:

- Диалект PostgreSQL языка SQL. В данном проекте используются сгенерированные запросы по вставке данных в систему;
- Java. Современный объектно-ориентированный язык программирования, на котором написана основа парсера. С помощью сторонних библиотек функциональные возможности языка значительно расширяются, открывая новые возможности внедрения бизнес-логики.

Для реализации проекта было использовано следующее программное обеспечение:

- pgAdmin 4. Система управления базами данных с графическим интерфейсом на основе PostgreSQL. Была создана база данных «schedule», в которую скрипт загружает считанные данные;
- IntelliJ Idea Professional Edition. Среда разработки на языке Java, позволяющая создавать рабочие проекты и подключать к ним различные модули, а также значительно оптимизирующая процесс разработки.

### 3. Реализация логики

Для реализации парсера были использованы следующие библиотеки:

- Apache Poi – библиотека, которая позволяет использовать файлы MS Office в Java приложениях.
- Spring Data JPA – модуль библиотеки Spring, предназначенный для подключения приложения к базе данных.
- Hibernate – библиотека для языка программирования Java, предназначенная для решения задач объектно-реляционного отображения.
- Lombok – небольшая аннотационная библиотека, позволяющая сократить количество шаблонного кода.
- Hibernate – библиотека, отвечающая за маппинг Java объектов к соответствующим таблицам в базе данных.

Разработанный парсер взаимодействует с реляционной базой данных PostgreSQL посредством готовых SQL-запросов типа INSERT.

Цикл работы скрипта во время работы с документом следует разделить на три ключевых этапа:

- Модуль, инициализирующий подключение к .xlsx таблице и подключение к базе данных;
- Модуль, считывающий данные из .xlsx таблиц и преобразующий их в объекты языка Java;
- Модуль, выполняющий транзакцию и вносящий полученные данные в PostgreSQL.

Первый этап представлен следующими функциями в файле DBConnection.java:

- getConnection() – функция, инициализирующая подключение к реляционной базе данных PostgreSQL, возвращает connection;
- connectXlsx() – функция, инициализирующая подключение к .xlsx таблице через глобальный путь к файлу, возвращает книгу в формате объекта Java.

Второй этап представлен функциями, взаимодействующими с Excel таблицами. Функции находятся в файле Parser.java:

- parseCompetentions() – считывает данные листа «Компетенции» выбранной .xlsx книги, возвращает список компетенций в формате объектов Java;
- parsePlan() – считывает данные листа «ПланСвод» выбранной .xlsx книги, возвращает список планов в формате объектов Java;
- parseTitular() – считывает данные листа «Титул» выбранной .xlsx книги, возвращает информацию с титульного листа в формате объекта Java.

Третий этап представлен функцией `commitTransaction()` файла `Transaction.java`, которая собирает все полученные парсером данные и вносит их в соответствующие таблицы базы данных: `Plans`, `Titulars`, `Competentions`. Для этого создаются объекты типа `PreparedStatement`, содержащие в себе необходимые SQL-запросы. После выполнения запросов подключение к базе данных автоматически закрывается, и парсер переходит в состояние готовности к обработке следующего документа.

### **Заключение**

В данной статье был рассмотрен процесс разработки и внедрения парсера для создания и заполнения реляционной базы данных, предназначенной для хранения учебных планов факультета компьютерных наук Воронежского государственного университета. В настоящее время в проект добавлены основные функциональные возможности: подключение к книге Excel, считывание данных и их выгрузка в базу данных. В перспективе проект функциональные возможности проекта будут расширены: в частности, ключевым вектором для совершенствования проектам является улучшение пользовательского интерфейса, с целью повышения удобства использования парсера.

### **Список литературы**

1. Реализация проекта по созданию реляционной базы данных для Контрольно-счётной палаты Воронежской области / А. Г. Потапов, А. В. Порядин, Т. Д. Братышев [и др.] // Труды молодых ученых факультета компьютерных наук ВГУ. Том Выпуск 2. – Воронеж : ООО «ВЭЛБОРН», 2022. – С. 132-137.
2. Разработка парсера для автоматизации деятельности Контрольно-счетной палаты Воронежской области / А. В. Порядин, А. Г. Потапов, Т. Д. Братышев [и др.] // Труды молодых ученых факультета компьютерных наук ВГУ. Том Выпуск 2. – Воронеж : ООО «ВЭЛБОРН», 2022. – С. 127-131.
3. Вариант решения задачи оптимизации толщины теплоизоляционного слоя в теплоснабжении / Н. А. Петрикеева, Д. М. Чудинов, Л. П. Мышовская // Научный журнал строительства и архитектуры. – 2018. – № 2(50). – С. 21-28.
4. Подольский, К. Д. Формирование предварительного коммерческого предложения ООО «ЧерноземАгроماش» средствами MS Excel / К. Д. Подольский, Я. Э. Головин, Е. А. Копытина // Сборник студенческих научных работ факультета компьютерных наук ВГУ : Сборник научных работ. В 2-х частях / Под редакцией Д.Н. Борисова.

Том Выпуск 14. Часть 1. – Воронеж : Воронежский государственный университет, 2020. – С. 143-147.

5. Кропачев, А. Н. Формирование отчета деятельности наркоконтроля по Воронежской области средствами офисных приложений / А. Н. Кропачев, Е. А. Копытина // Труды молодых учёных факультета компьютерных наук ВГУ : Сборник статей / Под редакцией Д.Н. Борисова. Том Выпуск 1. – Воронеж : Общество с ограниченной ответственностью "Вэлборн", 2021. – С. 80-84.

6. Копытин, А. В. О распределении штрафов и вознаграждений в проектах / А. В. Копытин, Д. И. Соломатин // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. – 2016. – № 4. – С. 119-126.

7. Васянкин, А. А. Модификация понятия «электронный документ» в системе электронного документооборота / А. А. Васянкин, Э. В. Кузьмина // Кайгородовские чтения : Материалы региональной научно-практической конференции, Краснодар, 29 апреля 2010 года / Краснодарский государственный университет культуры и искусств. – Краснодар: Краснодарский государственный университет культуры и искусств, 2010. – С. 75-79.

8. Application of game theory when taking decisions and choosing the best variant / N. Petrikeeve, Y. Kopytina, G. Martynenko, V. Kozlova // E3S Web of Conferences : Topical Problems of Green Architecture, Civil and Environmental Engineering, TPACEE 2019, Moscow, 20–22 ноября 2019 года. Vol. 164. – Moscow: EDP Sciences, 2020. – P. 08027. – DOI 10.1051/e3sconf/202016408027.