DOI: 10.12731/2227-930X-2025-15-1-351 **EDN: HLPOVZ**
УДК 004.415.25

Научная статья |
Системный анализ, управление и обработка информации, статистика

РАЗРАБОТКА ПРОГРАММЫ ШАБЛОНИЗАТОРА ДЛЯ АВТОМАТИЗАЦИИ УПРАВЛЕНИЯ ДАННЫМИ

Г.А. Гареева, А.Г. Файзуллина, А.А. Мирсаитова, З.Ш. Аглямова, Н.Р. Закирова

Аннотация

Обоснование. Интегрированные среды разработки, имеют ограниченные возможности для автоматизации повторяющихся задач, что приводит к росту временных затрат на разработку. Создание программного шаблонизатора решает эту проблему, автоматизируя рутинные процессы и снижая ошибки.

Цель – разработка программы, обеспечивающей автоматизацию создания и редактирования файлов шаблонов, их интеграцию с базой данных, а также поддержку просмотра содержимого файлов.

Метод и методология. Использование языка программирования Java для реализации программы и PostgreSQL для хранения данных. Подход основывался на применении универсальных инструментов для обеспечения совместимости и масштабируемости.

Результаты. В данной статье подробно рассмотрена разработка программы шаблонизатора, которая поддерживает создание, редактирование, просмотр и обновление файлов шаблонов. Шаблоны хранятся в PostgreSQL, что обеспечивает централизованное управление данными. Программа поддерживает принцип модульности, обеспечивающий расширение функционала программы за счет добавления необходимых флагов. При этом программный код будет требовать минимального вмешательства по причине внесенных изменений.

Область применения. Решение может быть использовано в промышленной автоматизации, образовательных стендах и при разработке проектов, требующих стандартизации процессов.

Выводы. Разработанная программа значительно сокращает время разработки и повышает точность выполнения повторяющихся операций при создании программ в интегрированной среде разработки для программируемых логических контроллеров.

Ключевые слова: автоматизация; шаблон; KincoBuilder; Java; PostgreSQL; программируемые логические контроллеры; промышленная автоматизация

Для цитирования. Гареева, Г. А., Файзуллина, А. Г., Мирсаитова, А. А., Аглямова, З. Ш., & Закирова, Н. Р. (2025). Разработка программы шаблонизатора для автоматизации управления данными. *International Journal of Advanced Studies*, *15*(1), 111–131. https://doi.org/10.12731/2227-930X-2025-15-1-351

Original article |
System Analysis, Information Management and Processing, Statistics

DEVELOPMENT OF A TEMPLATING PROGRAM FOR DATA MANAGEMENT AUTOMATION

G.A. Gareeva, A.G. Faizullina, A.A. Mirsaitova, Z.Sh. Aglyamova, N.R. Zakirova

Abstract

Background. Integrated development environments, have limited ability to automate repetitive tasks, which leads to increased development time. Creating a software templating tool solves this problem by automating routine processes and reducing errors.

Purpose. Development of a program providing automation of creation and editing of template files, their integration with the database, as well as support for viewing the content of the files.

Method and methodology. Using Java programming language for program implementation and PostgreSQL for data storage. The approach was based on the use of universal tools to ensure compatibility and scalability.

Results. This paper details the development of a templateizer program that supports creating, editing, viewing, and updating template files. Templates are stored in PostgreSQL, which provides centralized data management. The program supports the principle of modularity, which ensures the expansion of the program functionality by adding the necessary flags. In this case, the program code will require minimal intervention due to the changes made.

Scope of the results. The solution can be used in industrial automation, educational testbeds and in the development of projects that require standardization of processes.

Conclusions. The developed program significantly reduces the development time and increases the accuracy of repetitive operations when creating programs in the integrated development environment for programmable logic controllers.

Keywords: automation; template; KincoBuilder; Java; PostgreSQL; programmable logic controllers; industrial automation

For citation. Gareeva, G. A., Faizullina, A. G., Mirsaitova, A. A., Aglyamova, Z. Sh., & Zakirova, N. R. (2025). Development of a templating program for data management automation. *International Journal of Advanced Studies*, *15*(1), 111–131. https://doi.org/10.12731/2227-930X-2025-15-1-351

Введение

Проблема автоматизации рутинных операций в интегрированных средах разработки (IDE) актуальна для многих сфер промышленной автоматизации. Программы, являющиеся специализированными средами для программируемых логических контроллеров (ПЛК), обладают ограниченными возможностями автоматизации задач, таких как управление шаблонами файлов. Решение данной проблемы требует создания универсального инструмента, способного интегрироваться с существующими системами разработки [7].

Цель и задачи

Целью работы является создание программы, способной автоматизировать создание, редактирование, просмотр и хранение шаблонов файлов для повышения эффективности работы в KincoBuilder.

KincoBuilder – интегрированная среда разработки, назначение которой является создание логических программ, обеспечивающих решение задач управления технологическим процессом на базе ПЛК Kinco Automation [9].

Для достижения цели решены следующие *задачи*:

- разработан пользовательский интерфейс для взаимодействия с файлами;
- реализована интеграция с базой данных PostgreSQL для централизованного хранения шаблонов;
- обеспечена возможность многострочного редактирования и просмотра содержимого файлов [13].

Материалы и методы разработки

Процесс разработки включал:

- Выбор инструментов. Использование Java для разработки программы и PostgreSQL для управления данными [4];
- Архитектура программы. Реализация командной строки для интерактивного взаимодействия с пользователем [15];
- Интеграция с базой данных. Использование JDBC для взаимодействия с PostgreSQL [5].

Реализация программы

Реализация программы включала в себя множество этапов, направленных на создание универсального инструмента для работы с шаблонами файлов в интегрированной среде разработки KincoBuilder [5].

Каждый из этапов был тщательно проработан с учётом требований к автоматизации, удобству использования и масштабируемости:

1. Анализ файлов, создаваемых программой KincoBuilder. Одним из первых шагов был анализ структуры файлов, генерируемых программой KincoBuilder [8].

Эти файлы содержат инструкции, необходимые для работы программируемых логических контроллеров. Пример такого файла представлен на рисунке 1.

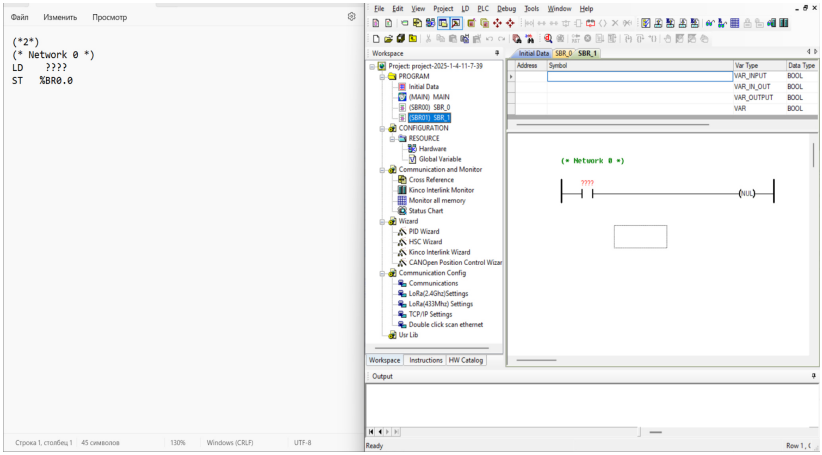

Рис. 1. Генерация текста-кода программой KincoBuilder при создании новой сети

На основе анализа было выделено несколько типовых шаблонов, которые могли бы быть использованы для автоматизации процессов разработки. Шаблоны включали стандартные конструкции программ, были добавлены шаблоны по созданию сетей (нетворков) и определения массивов. Сети (Нетворки) – это логический блок программы, представляющий собой последовательность инструкций для выполнения определённой операции или группы операций. Результаты анализа позволили определить структуру шаблонов, которая будет храниться в базе данных. Для каждого шаблона была предусмотрена возможность интеграции в существующие файлы KincoBuilder, что обеспечивает его удобное использование в реальных проектах.

2. Настройка инструментов разработки. Для реализации программы использовался язык программирования Java благодаря его универсальности, богатой экосистеме библиотек и поддержке работы с базами данных через JDBC [11].

Для хранения данных о шаблонах была выбрана PostgreSQL, предоставляющая надёжность и мощные инструменты для работы с текстовыми данными. Среда разработки была настроена таким образом, чтобы обеспечивать непрерывный процесс тестирования и отладки. Использовались библиотека Apache Commons CLI для обработки командной строки, а также встроенные средства логирования Java для мониторинга выполнения программы [10].

На рисунке 2 представлено окно интегрированной среды разработки Intellij IDEA, которое включает в себя содержимое подключаемых зависимостей в файле pom.xml, связанные со сборщиком проекта.

Зависимости – это библиотеки или модули, которые проект использует для работы. Например, если нужно подключить базу данных, необходимо добавить зависимость — библиотеку для работы с этой базой. Maven автоматически скачает её и всё, что ей нужно, чтобы работать (транзитивные зависимости) [15].

Сборка – это процесс подготовки проекта к использованию, включающий:

- компиляцию кода;
- упаковку в удобный формат (например, .jar или .war);
- тестирование для проверки ошибок. Сборщик проектов Maven делает это автоматически на основе настроек в pom.xml.

Файл pom.xml (Project Object Model) – это главный файл конфигурации в проектах, которые используют систему сборки Apache Maven. Он представляет собой XML-документ, содержащий информацию о проекте, его зависимостях, настройках сборки, плагинах и других аспектах, необходимых для управления жизненным циклом проекта [3].

Файл pom.xml важен для:

- Автоматизация. Maven упрощает процесс сборки, тестирования и упаковки;
- Управление зависимостями. Не нужно вручную скачивать библиотеки;
- Транзитивность. Maven автоматически добавляет зависимости, требуемые подключаемыми библиотеками;
- Консистентность. Проект можно легко настроить для различных сред с использованием профилей.

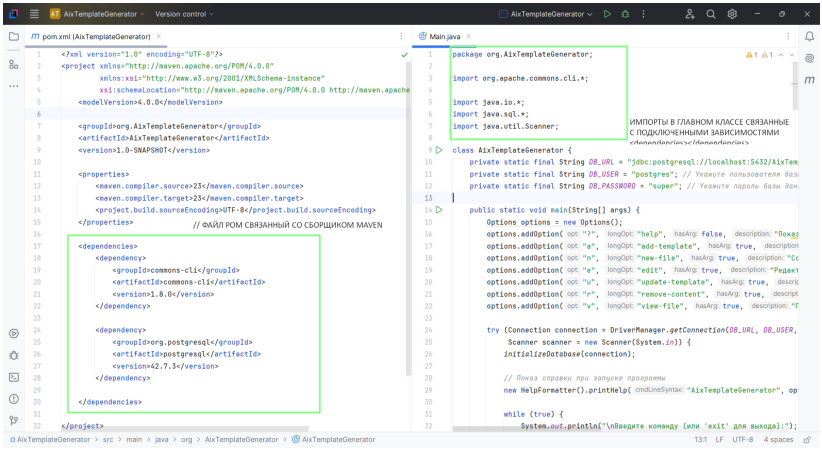

Рис. 2. Зависимости, подключаемые в проект

3. Заполнение шаблонов в базу данных показано на рисунке 3. База данных (БД) была спроектирована для хранения шаблонов, включая их уникальные идентификаторы, названия и содержимое.

Это позволило обеспечить лёгкий доступ к шаблонам для их редактирования, обновления и последующего использования. Каждый шаблон можно вводить в базу данных напрямую через запросы или с помощью команды, позволяющей пользователю загрузить содержимое в систему. Это делает процесс управления шаблонами централизованным и организованным.

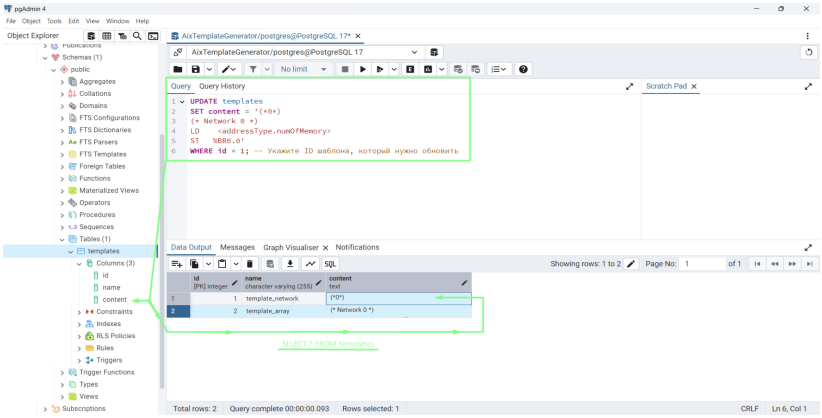

Рис. 3. Заполнение таблицы templates необходимыми шаблонами непосредственно с помощью запросов к БД

На рисунке 4 представлено заполнение шаблоном БД непосредственно со стороны пользователя с помощью терминала. Терминал – это окно, где вводятся команды (например, запустить программу, создать файл, скопировать папку) [12].

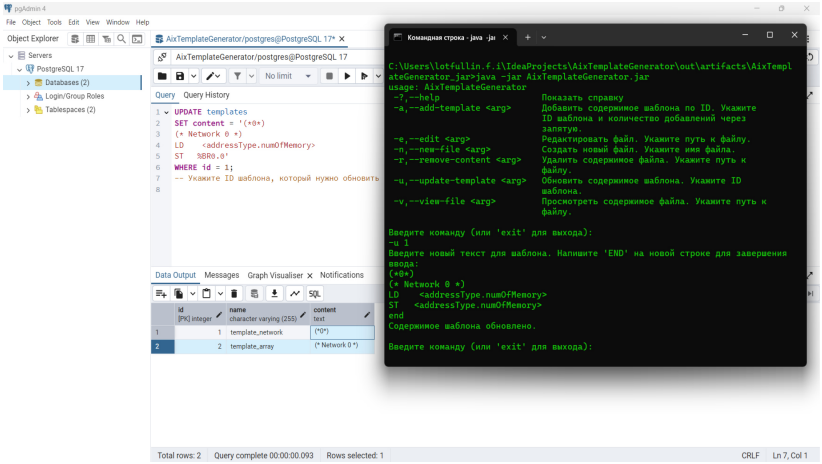

Рис. 4. Обновление – заполнение таблицы templates необходимыми шаблонами с помощью флага -u непосредственно с помощью терминала, являющегося частью консольного интерфейса пользователя

4. Синхронизация с базой данных. Система была разработана таким образом, чтобы обеспечивать синхронизацию между локальными файлами и базой данных [1].

При создании или редактировании файла информация автоматически обновлялась в базе данных, а внесённые изменения сразу же становились доступны для последующего использования. Синхронизация позволила избежать дублирования данных и повысила надёжность хранения. Кроме того, это сделало возможным совместную работу нескольких разработчиков с единым набором шаблонов.

На рисунке 5 представлен процесс обмена и обработки данных между интерфейсом пользователя, программой и базой данных.

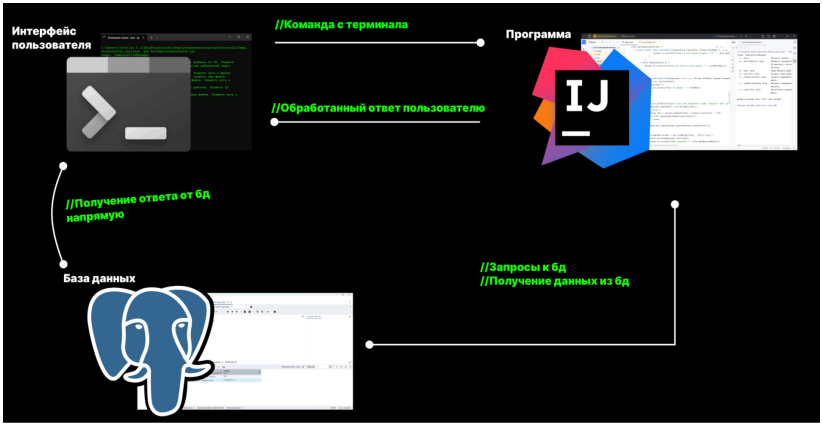

Рис. 5. Процесс обработки программных данных

5. Написание кода программы и тестирование. Процесс написания кода включал реализацию функциональных модулей программы [2]. Взаимодействие с программой сделано с длинными и короткими флагами (- и --). Преимущества данного решения включают:

- Гибкость. Позволяют одной команде выполнять множество разных задач.

- Удобство. Настраивают поведение программы без изменения её кода.

В данную программу были добавлены следующие команды:

- Обновление шаблона. Программа предоставляет возможность редактировать шаблоны, хранящиеся в базе данных, с поддержкой многострочного ввода. После внесения изменений они сохраняются в системе. Данная команда была показана на рисунке 3 с использованием флага -u.
- Создание файла. Создаётся новый текстовый файл, который регистрируется в базе данных. Пользователь может указать название файла, после чего он будет автоматически заполнен шаблонным содержимым. Данная команда представлена на рисунке 6.

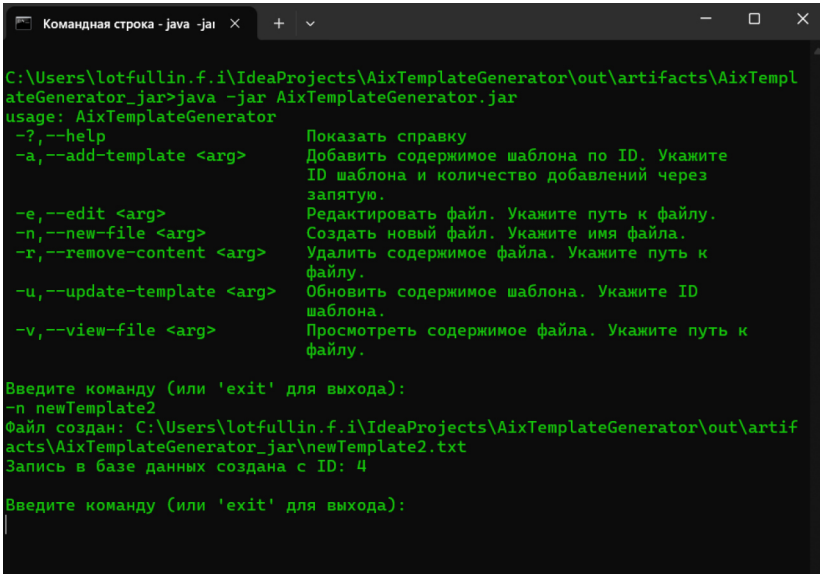

Рис. 6. Процесс создания файла с использованием флага -n

- Редактирование файла. Программа позволяет добавлять многострочный текст в файл, поддерживая завершение вво-

да через ключевое слово END. Это удобно для внесения изменений в файл без необходимости использования сторонних редакторов. Данная команда показана на рисунке 7.

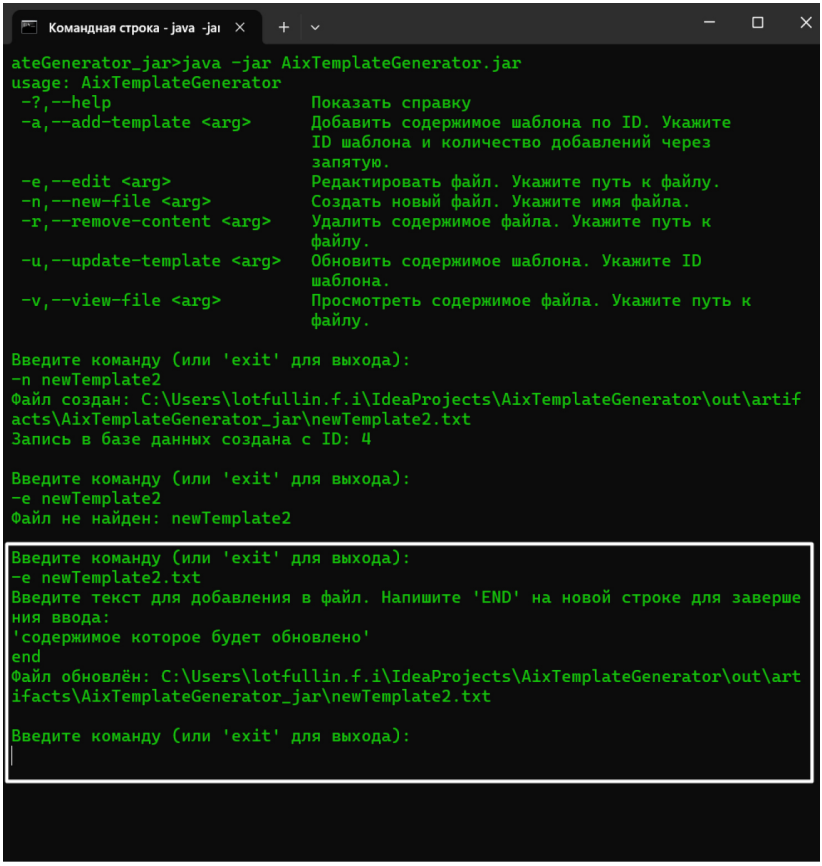

Рис. 7. Процесс редактирования файла с использованием флага -e

- Просмотр содержимого. Пользователь может просматривать содержимое файла прямо из программы. Это ускоряет процесс работы и позволяет быстро проверить правильность данных. Данная команда представлена на рисунке 8.

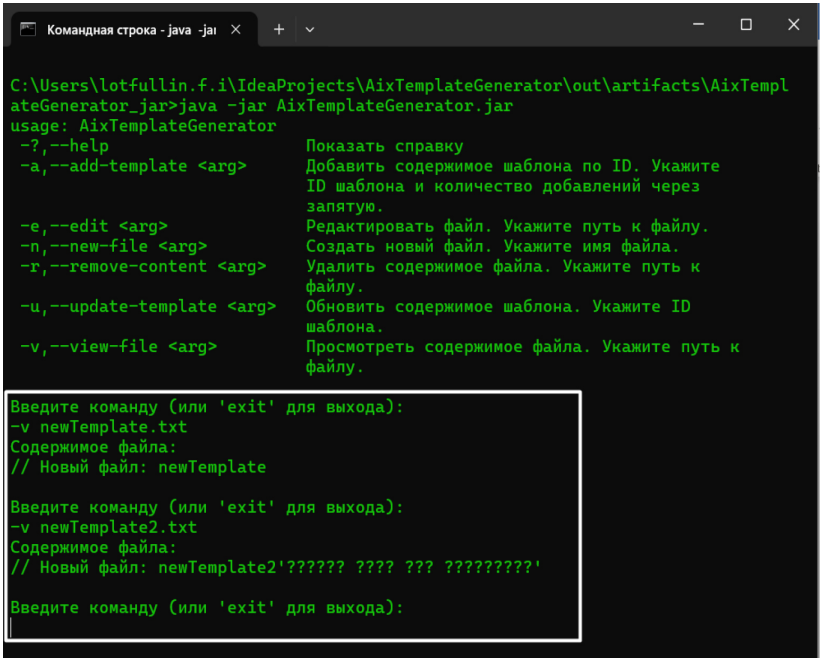

Рис. 8. Просмотр содержимого файла с использованием флага -v

- Удаление содержимого. При необходимости содержимое файла может быть полностью очищено, оставляя только структуру. Данная команда представлена на рисунке 9.

После написания каждой функциональности проводилось тестирование, чтобы убедиться в её корректной работе [14]. Тесты включали проверку работы с различными типами шаблонов, ситуациями с отсутствием данных и некорректным вводом. Это позволило устранить большинство ошибок до выпуска программы.

В качестве примера тестирования представлено внедрение шаблонов в существующие программные файлы Kincobuilder. На рисунке 10 показано внедрение пользователем своих шаблонов в программные файлы программы Kincobuilder с расширением файлов .ilp.

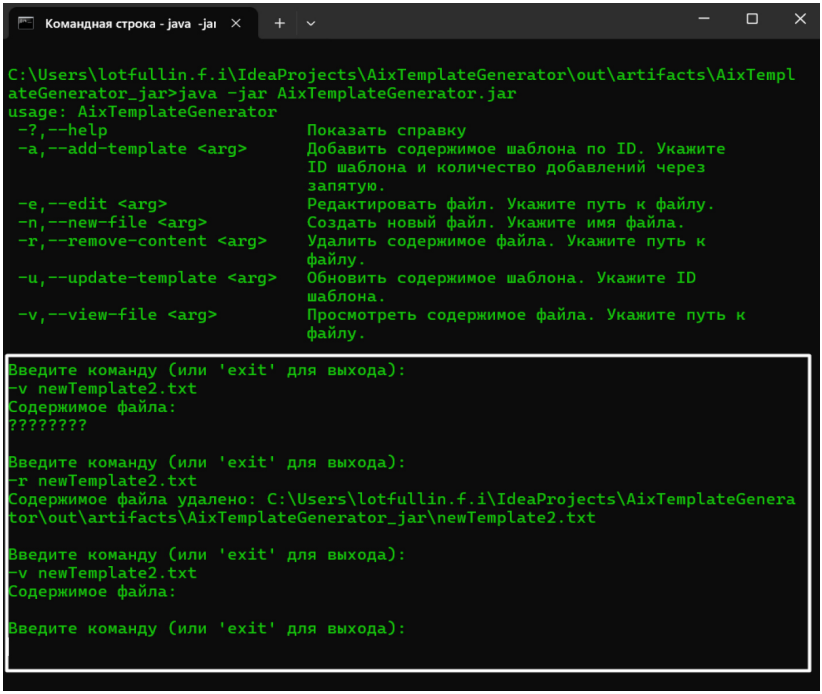

Рис. 9. Удаление содержимого файла с использованием флага -r

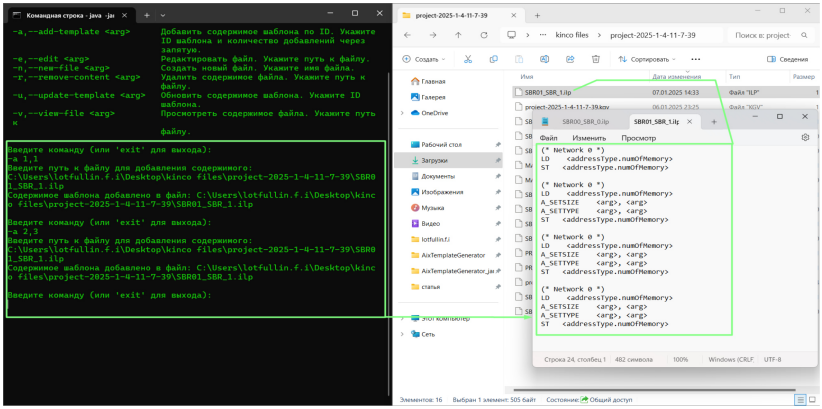

Рис. 10. Внедрение пользователем своих шаблонов кода в программные файлы, создаваемые средой разработки Kincobuilder

На рисунке 11 показан результат внедрения шаблонов кода в программные файлы программы Kincobuilder. В интегрированной среде разработки Kincobuilder создаются сети с логикой, которые были внедрены пользователем.

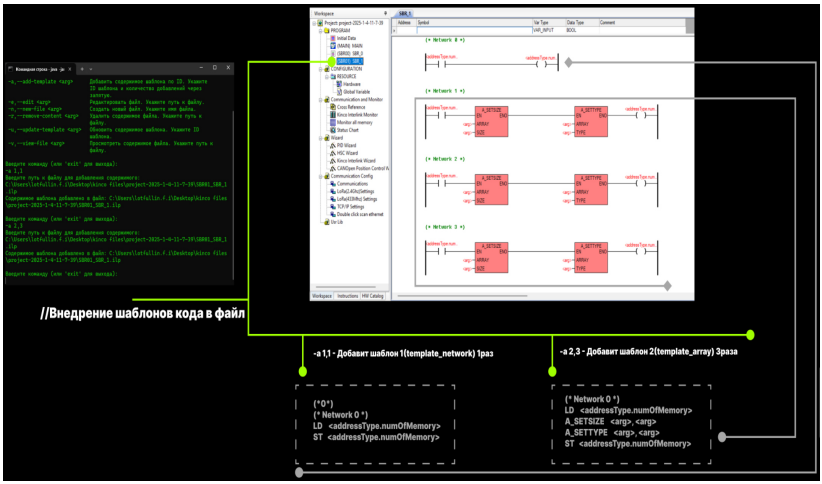

Рис. 11. Результат внедрения программных кодов в файлы Kincobuilder

Программа была структурирована для обеспечения удобства использования и лёгкости расширения. Например, добавление новых команд или поддержка дополнительных типов шаблонов может быть реализовано с минимальными изменениями в коде.

Таким образом, реализация программы охватывала весь спектр задач, связанных с автоматизацией рутинных процессов в KincoBuilder, и обеспечила удобный интерфейс для разработчиков.

Область применения

Программа может использоваться в следующих целях:

- в промышленной автоматизации - для стандартизации работы с шаблонами файлов;
- в образовательных целях - при обучении работе с шаблонами файлов и базами данных;

- в разработке ПО - для ускорения процесса создания однотипных программ.

Заключение

Созданная программа решает проблему автоматизации рутинных операций в KincoBuilder. Использование открытых инструментов, таких как Java и PostgreSQL, обеспечивает гибкость и расширяемость решения. Дальнейшее развитие может включать поддержку графического интерфейса пользователя, который позволяет взаимодействовать с программой через визуальные элементы, такие как окна, кнопки, меню и иконки для повышения удобства пользователей.

Список литературы

1. Бондаренко, И. С. (2019). *Базы данных: создание баз данных в среде SQL Server*. Лабораторный практикум. Москва: Изд. Дом НИТУ «МИСиС». 39 с.
2. Линец, Г. И., & Братченко, Н. Ю. (2021). *Базы данных*. Учебник. Ставрополь: Изд-во СКФУ. 170 с.
3. Фиайли, К. (2023). *SQL. Руководство для использования с любыми SQL СУБД*. Учебное пособие (2-е изд.; пер. с англ. А. В. Хаванова). Москва: ДМК Пресс. 454 с.
4. Гайнанова, Р. Ш. (2019). *Разработка приложений в Visual C для работы с базой данных MS SQL SERVER 2012*. Учебно-методическое пособие. Казань: КНИТУ. 84 с.
5. Ёсу, М. (2021). *Принципы организации распределенных баз данных*. Учебник (пер. с англ. А. А. Слинкина). Москва: ДМК Пресс. 672 с.
6. Харрингтон, Д. (2023). *Проектирование объектно-ориентированных баз данных*. Практическое руководство (2-е изд.; пер. с англ. А. А. Слинкина). Москва: ДМК Пресс. 273 с.
7. Льюис, Д. (2023). *Ядро Oracle. Внутреннее устройство для администраторов и разработчиков баз данных*. Практическое ру-

ководство (2-е изд.; пер. с англ. А. Н. Киселева). Москва: ДМК Пресс. 373 с.

8. Мартишин, С. А., Симонов, В. Л., & Храпченко, М. В. (2024). *Базы данных. Практическое применение СУБД SQL и NoSQL типа для проектирования информационных систем*. Учебное пособие. Москва: ФОРУМ: ИНФРА-М. 368 с.
9. Мартишин, С. А., Симонов, В. Л., & Храпченко, М. В. (2022). *Базы данных: проектирование и разработка информационных систем с использованием СУБД MySQL и языка Go*. Учебное пособие. Москва: ИНФРА-М. 325 с. https://doi.org/10.12737/1830834
10. Митин, А. И. (2020). *Работа с базами данных Microsoft SQL Server: сценарии практических занятий*. Учебно-методическое пособие. Москва: Директ-Медиа. 143 с.
11. Amin Al Ka'bi. (2021). Management of energy consumption using programmable logic controllers (PLCs). *Proceedings on Engineering Sciences*, *3*(3), 267–272. https://doi.org/10.24874/pes03.03.003
12. Walters III, E. G., & Bryla, E. J. (2016). The impact of PLC program architecture on production line efficiency: Case study of a control system rewrite. *Machines*, *4*(2), 13. https://doi.org/10.3390/machines4020013
13. Martin A. Sehr et al. (2024). Programmable Logic Controllers in the context of industry 4.0. *IEEE Journals & Magazine*. Получено с https://ieeexplore.ieee.org/document/9134804
14. Tiago Cruz et al. (2024). Virtualizing Programmable Logic Controllers: toward a convergent approach. *IEEE Journals & Magazine*. Получено с https://ieeexplore.ieee.org/document/7564414
15. Zheng Yang et al. (2021). PLCrypto: A symmetric cryptographic library for programmable logic controllers. *IACR Transactions on Symmetric Cryptology*, *2021*(3), 170–217. https://doi.org/10.46586/tosc.v2021.i3.170-217

References

1. Bondarenko, I. S. (2019). Databases: Creating databases in SQL Server environment. Laboratory manual. Moscow: MISiS Publishing House. 39 p.

2. Linec, G. I., & Bratchenko, N. Y. (2021). *Databases*. Stavropol: SKFU Publishing House. 170 pp.
3. Fiayli, K. (2023). *SQL: User Guide for Any SQL DBMS* (2nd ed.; translated by A. V. Khavanov). Moscow: DMK Press. 454 p.
4. Gainanova, R. Sh. (2019). *Application Development in Visual C++ for Working with MS SQL Server 2012 Database*. Tutorial. Kazan: KNITU. 84 p.
5. Yo, M. (2021). *Principles of Distributed Database Organization* (translated by A. A. Slinkin). Moscow: DMK Press. 672 p.
6. Harrington, D. (2023). *Object-Oriented Database Design: Practical Guide* (2nd ed.; translated by A. A. Slinkin). Moscow: DMK Press. 273 p.
7. Lewis, D. (2023). *Oracle Kernel: Internal Workings for DBAs and Developers* (2nd ed.; translated by A. N. Kiselev). Moscow: DMK Press. 373 p.
8. Martishin, S. A., Simonov, V. L., & Khrapchenko, M. V. (2024). *Databases: Practical Application of SQL and NoSQL DBMS for Information Systems Design*. Moscow: Forum INFRA-M. 368 p.
9. Martishin, S. A., Simonov, V. L., & Khrapchenko, M. V. (2022). *Databases: Design and Development of Information Systems Using MySQL DBMS and Go Programming Language*. Moscow: INFRA-M. 325 pp. https://doi.org/10.12737/1830834
10. Mitin, A. I. (2020). *Working with Microsoft SQL Server Databases: Practical Exercise Scenarios*. Moscow: Direct-Media. 143 p.
11. Amin Al Ka'bi. (2021). Management of energy consumption using programmable logic controllers (PLCs). *Proceedings on Engineering Sciences*, *3*(3), 267–272. https://doi.org/10.24874/pes03.03.003
12. Walters III, E. G., & Bryla, E. J. (2016). The impact of PLC program architecture on production line efficiency: Case study of a control system rewrite. *Machines*, *4*(2), 13. https://doi.org/10.3390/machines4020013
13. Martin A. Sehr et al. (2024). Programmable Logic Controllers in the context of Industry 4.0. *IEEE Journals & Magazine*. Retrieved from https://ieeexplore.ieee.org/document/9134804

14. Tiago Cruz et al. (2024). Virtualizing Programmable Logic Controllers: toward a convergent approach. *IEEE Journals & Magazine*. Retrieved from https://ieeexplore.ieee.org/document/7564414
15. Zheng Yang et al. (2021). PLCrypto: A symmetric cryptographic library for programmable logic controllers. *IACR Transactions on Symmetric Cryptology*, *2021*(3), 170–217. https://doi.org/10.46586/tosc.v2021.i3.170-217

ВКЛАД АВТОРОВ

Гареева Г.А.: разработка программы шаблонизатора для автоматизации управления данными, обработка результатов исследований.

Файзуллина А.Г.: формулирование основных направлений исследования, разработка теоретических предпосылок, формирование общих выводов.

Мирсаитова А.А.: проведение сбора данных, подготовка начального варианта статьи.

Аглямова З.Ш.: научное редактирование текста статьи и окончательное утверждение версии для публикации.

Закирова Н.Р.: анализ и интерпретация полученных данных, литературный анализ.

AUTHORS CONTRIBUTION

Gulnara A. Gareeva: development of the templateizer program for automation of data management, processing of research results.

Aigul G. Faizullina: formulation of the main directions of the research, development of theoretical assumptions, formation of general conclusions.

Asiya A. Mirsaitova: carrying out data collection, preparation of the initial version of the article.

Zulfina Sh. Aglyamova: scientific editing of the text of the article and final approval of the version for publication.

Nuriya R. Zakirova: analysis and interpretation of the obtained data, literary analysis.

ДАННЫЕ ОБ АВТОРАХ

Гареева Гульнара Альбертовна, кандидат педагогических наук, доцент, заведующий кафедрой информационных систем
Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ
ул. Академика Королева, 1, г. Набережные Челны, 423814, Российская Федерация
gagareeva1977@mail.ru

Файзуллина Айгуль Гинатулловна, старший преподаватель кафедры бизнес-информатики и математических методов в экономике
Казанский федеральный университет Набережночелнинский институт
проспект Мира, 68/19, г. Набережные Челны 423812, Российская Федерация
dlya_pisem_t@mail.ru

Мирсаитова Асия Акмалетдиновна, старший преподаватель кафедры высшей математики и информационных технологий
Казанский инновационный университет им. В.Г. Тимирясова
пр. Московский, 67, г. Набережные Челны, 423822, Российская Федерация
a230864m@yandex.ru

Аглямова Зульфина Шамилевна, кандидат педагогических наук, доцент кафедры высшей математики, моделирования и анализа данных
Казанский инновационный университет им. В.Г. Тимирясова
пр. Московский, 67, г. Набережные Челны, 423822, Российская Федерация
aglamova@chl.ieml.ru

Закирова Нурия Ришатовна, кандидат педагогических наук, доцент кафедры информатики и вычислительной математики
Набережночелнинский государственный педагогический университет
ул. Им. Низаметдинова Р.М., 28, г. Набережные Челны, 423806, Российская Федерация
smile-nuriya@yandex.ru

DATA ABOUT THE AUTHORS

Gulnara A. Gareeva, Candidate of Pedagogical sciences, Associate professor, Head of the Department of Information Systems
Kazan National Research Technical University named after A.N. Tupolev-KAI
1, Akademika Koroleva Str., Naberezhnye Chelny, 423814, Russian Federation
gagareeva1977@mail.ru
SPIN-code: 3279-8465
Scopus Author ID: 36801593200
ResearcherID: M-1728-2015
ORCID: https://orcid.org/0000-0002-8539-4541

Aigul G. Faizullina, Senior Lecturer of the Department of Business Informatics and Mathematical Methods in Economics
Kazan Federal University Naberezhnochelninsk Institute
68/19 Prospekt Mira, Naberezhnye Chelny 423812, Russian Federation
dlya_pisem_t@mail.ru

Asiya A. Mirsaitova, Senior Lecturer, Department of Higher Mathematics and Information Technologies
Kazan Innovation University named after V.G. Timiryasov
67, Moskovsky Ave., Naberezhnye Chelny, 423822, Russian Federation
a230864m@yandex.ru

Zulfina Sh. Aglyamova, Candidate of Pedagogical Sciences, Associate Professor of the Department of Higher Mathematics, Modeling and Data Analysis
Kazan Innovation University named after V.G. Timiryasov
67, Moskovsky Ave., Naberezhnye Chelny, 423822, Russian Federation
aglamova@chl.ieml.ru

Nuriya R. Zakirova, Candidate of Pedagogical Sciences, Associate Professor, Department of Informatics and Computational Mathematics
Naberezhnochelninsk State Pedagogical University
Im. Nizametdinov R.M. Str., 28, Naberezhnye Chelny, 423806, Russian Federation
smile-nuriya@yandex.ru

Поступила 05.02.2025
После рецензирования 01.03.2025
Принята 03.03.2025

Received 05.02.2025
Revised 01.03.2025
Accepted 03.03.2025