Министерство просвещения Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«Российский государственный профессионально-педагогический университет»

Институт инженерно-педагогического образования

Кафедра информационных систем и технологий

РАЗРАБОТКА WINDOWS ПРИЛОЖЕНИЯ ДЛЯ РАБОТЫ С ВНЕШНИМИ ИСТОЧНИКАМИ ДАННЫХ (БД) НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ JAVA

Курсовая работа

По дисциплине «Технологии программирования»

Студент группы Пим-201

Бирюков Д.В.

Руководитель:

Ассистент кафедры ИС

Шевцов А. С.

Екатеринбург 2023 г.

Оглавление

[1. Введение. 4](#_Toc16585)

[1.1 Описание предметной области. 4](#_Toc26243)

[1.2 Цель работы. 4](#_Toc3993)

[1.3 Задачи исследования. 4](#_Toc29943)

[1.4 Обоснование актуальности выбранной темы. 5](#_Toc21677)

[2. Обзор литературы. 6](#_Toc12560)

[2.1 Обзор JavaFX и его возможностей для разработки графического интерфейса. 6](#_Toc26756)

[2.2 Исследование существующих решений для отслеживания расписания занятий и добавления заметок. 6](#_Toc24256)

[2.3 Использование API запросов для получения расписания занятий из внешних источников. 6](#_Toc24312)

[2.4 Исследование существующих решений для отслеживания, выявление и исправление ошибок и недочётов. 7](#_Toc19320)

[3. Анализ требований и проектирование. 8](#_Toc15532)

[3.1 Определение функциональных и нефункциональных требований к приложению. 8](#_Toc2664)

[3.2 Проектирование архитектуры приложения, включая модели данных, контроллеры и представления . 8](#_Toc11485)

[3.3 Разработка схемы базы данных для хранения расписания занятий и заметок. 8](#_Toc4983)

[3.4 Описание пользовательского интерфейса приложения. 10](#_Toc25324)

[4. Реализация 11](#_Toc25393)

[4.1 Создание основного окна приложения с отображением расписания занятий. 11](#_Toc27937)

[4.2 Разработка функциональности добавления, вывода расписания, изменения персональных данных и добавления заметок. 11](#_Toc27399)

[4.3 Интеграция с базой данных для хранения расписания и заметок. 11](#_Toc10854)

[4.4 Взаимодействие с внешними источниками данных для получения расписания занятий. 12](#_Toc23687)

[5. Тестирование и отладка 13](#_Toc25367)

[5.1 Планирование и проведение тестирования приложения. 13](#_Toc28547)

[5.2 Выявление и исправление ошибок и недочётов. 13](#_Toc17688)

[5.3 Оптимизация производительности и устранение узких мест. 13](#_Toc573)

[6. Результаты и анализ . 14](#_Toc17870)

[6.1 Оценка выполнения требований и достижения целей работы. 14](#_Toc11276)

[6.2 Анализ эффективности разработанного приложения 14](#_Toc16561)

[7. Заключение 15](#_Toc24090)

[7.1 Основные выводы работы. 15](#_Toc8073)

[7.2 Резюме выполненной работы. 15](#_Toc19244)

[7.3 Рекомендации для дальнейшего развития и улучшения приложения. 16](#_Toc418)

[8. Список литературы 17](#_Toc22164)

[9. Приложение 18](#_Toc16751)

[9.1 Скриншоты интерфейса приложения 18](#_Toc3001)

# Введение.

## Описание предметной области.

В современном мире, где люди активно занимаются учёбой, работой и другими обязанностями, эффективное планирование и отслеживание расписания занятий становится все более важным. Студенты нуждаются в надёжном инструменте, который поможет им управлять своим временем, чтобы достичь максимальной продуктивности. В рамках данной работы будет разрабатываться JavaFX приложение, предназначенное для отслеживания расписания занятий и добавления заметок, чтобы помочь пользователям более эффективно организовывать свою деятельность.

## Цель работы.

Целью данной работы является разработка JavaFX приложения, которое позволит пользователям удобно отслеживать своё расписание занятий и добавлять заметки для каждого дня. Приложение должно обладать интуитивно понятным интерфейсом, функциональностью по отслеживанию расписания и добавлению заметок, а также интеграцией с внешними источниками данных для получения актуального расписания занятий.

## Задачи исследования.

Для достижения поставленной цели в рамках работы будут выполнены следующие задачи:

* Изучение возможностей JavaFX для разработки графического интерфейса приложения;
* Исследование методов взаимодействия с внешними источниками данных для получения расписания занятий;
* Разработка функциональности добавления, редактирования и удаления заметок в приложении;
* Интеграция с базой данных для хранения расписания занятий и заметок;
* Тестирование приложения и оценка его производительности.

## Обоснование актуальности выбранной темы.

Современный ритм жизни требует от людей эффективного управления своим временем и обязанностями. Отслеживание расписания занятий и добавление заметок играют важную роль в организации дня и достижении поставленных целей. Разработка JavaFX приложения для удобного отслеживания расписания и добавления заметок обладает практической значимостью, так как позволяет улучшить организацию времени и повысить продуктивность. Также данная тема актуальна с точки зрения применения современных технологий программирования и разработки пользовательских интерфейсов.

# Обзор литературы.

## Обзор JavaFX и его возможностей для разработки графического интерфейса.

JavaFX является современной технологией разработки графического интерфейса на языке Java. В литературе проведён обзор основных функций и возможностей JavaFX, таких как создание пользовательских элементов управления, работа с макетами, анимации, стилизация интерфейса и другие. Также рассмотрены принципы построения архитектуры JavaFX приложений и рекомендации по организации кода.

## Исследование существующих решений для отслеживания расписания занятий и добавления заметок.

В ходе литературного обзора были изучены существующие решения и приложения, которые предназначены для отслеживания расписания занятий и добавления заметок. Были проанализированы их функциональности, особенности интерфейса и способы интеграции с внешними источниками данных. Это позволило выделить ключевые аспекты, которые следует учесть при разработке собственного JavaFX приложения.

## Использование API запросов для получения расписания занятий из внешних источников.

В рамках литературного обзора было произведено исследование API запросов для получения расписания занятий из внешних источников данных. Были изучены протоколы и форматы данных, используемые для передачи расписания, а также методы и способы взаимодействия с API. Это позволило определить наиболее подходящие технологии и инструменты для интеграции в разрабатываемое JavaFX приложение.

## Исследование существующих решений для отслеживания, выявление и исправление ошибок и недочётов.

В ходе литературного обзора были изучены существующие решения и подходы для отслеживания, выявления и исправления ошибок и недочётов в разработке JavaFX приложений. Были рассмотрены различные методы тестирования, отладки и контроля качества кода. Это позволило определить наиболее эффективные практики и инструменты для обеспечения надёжности и стабильности разрабатываемого приложения.

# Анализ требований и проектирование.

## Определение функциональных и нефункциональных требований к приложению.

Базовыми функциями и требованием к моему приложению будет регистрация пользователя, возможность верификации, средне надёжное хранение данных пользователя. Возможность выбора своей учебной группы и подгруппы с возможностью дальнейшего изменения этих данных. Возможность выбора цветовой темы приложение и дальнейшее её изменение в зависимости от пользовательского предпочтения. Добавление списка задач на активные дни обучения, и возможность задавать себе задания на день.

## Проектирование архитектуры приложения, включая модели данных, контроллеры и представления .

Архитектура приложения будет состоять из шести элементов:

1. Главный класс благодаря которому и будет запускаться приложение.
2. FXML файлы разметки графической оболочки.
3. CSS файлы определяющие внешний вид элементов FXML.
4. Классы контроллеры для всех FXML файлов, которые будут отвечать за логику приложения.
5. База данных PostgreSQL в которой будут храниться данные.
6. Класс DataBaseHandler в котором будет находиться вся логика и связь контрейлеров и базы данных.

## Разработка схемы базы данных для хранения расписания занятий и заметок.

База данных будет разрабатываться в pgAdmin4 и будет состоять из девяти таблиц:

1. Словарь аудиторий.
2. Словарь расписания звонков для всех курсов.
3. Словарь существующих дисциплин.
4. Словарь всех учебных групп учебного заведения.
5. Словарь всех преподавателей университета.
6. Словарь всех типов занятий проводимых в университете.
7. Таблица объединяющая все словари, хранящая в себе расписание всех групп на две недели вперёд.
8. Таблица пользователей приложения хранящая в себя персональные данных пользователя.
9. Таблица задач пользователей хранящая в себе все активные задачи пользователей.

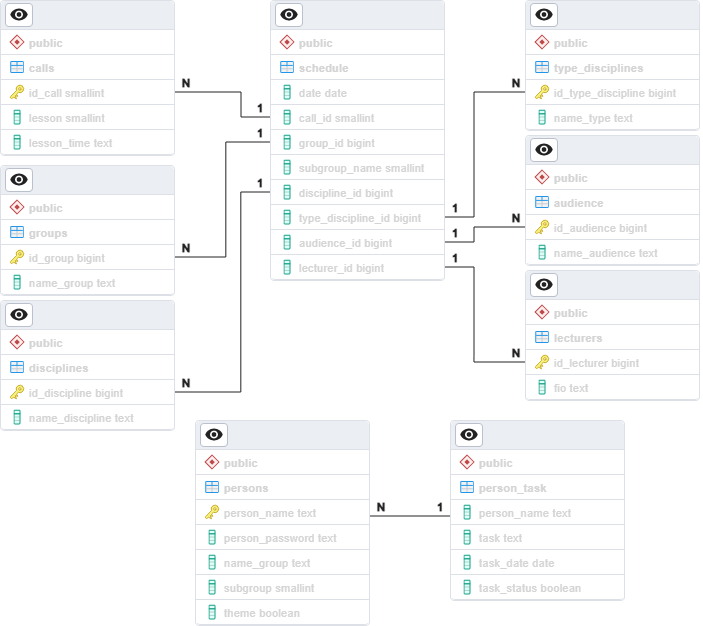


Рис 1. Даталогическая таблица базы данных.

## Описание пользовательского интерфейса приложения.

Пользовательский интерфейс состоит из трёх окон: окно регистрации входа и главное окно. Все они связанны между собой имеют приятный дизайн и небольшой размер.

# Реализация

## Создание основного окна приложения с отображением расписания занятий.

Создание основного окна приложения проходило в несколько этапов, первым этапом нужно было получить расписание, это вызвало большие затруднение, так как технология api в университете не идеальна и имеет изъяны. Вторым этапом нужно было создать окна регистрации и верификации так как вывод расписания возможен только с указанием учебной группы и подгруппы, после я приступил к созданию массивного главного окна имеющего множество функций.

## Разработка функциональности добавления, вывода расписания, изменения персональных данных и добавления заметок.

Функциональное наполнение проходило в несколько этапов, первичное наполнение, дальнейший анализ и оптимизация. Этот подход в отличии от едино разового чистового кода дал мне большие плоды, так как дал мне большую почву для размышлений и большую оптимизацию кода.

## Интеграция с базой данных для хранения расписания и заметок.

Работа с базой данных проходит из класса контроллера, путём создания экземпляра класса DataBaseHandler, а уже в нём я прописал едино разовое подключение к базе данных что дало прирост к скорости работы приложения и сделало код более читабельным. В это же классе находятся все функции работы с базой данных. Для всех нужных изменяемых данных из базы данных я создал отдельные методы и прописал обработчик ошибок в каждом что дало мне преимущество при тестировании приложения.

## Взаимодействие с внешними источниками данных для получения расписания занятий.

Единственным внешним источником взаимодействия была api структура вуза, отправляя запросы в которую я получал JSON массивы данных которые в дальнейшем и обрабатывал для получения расписания занятий.

# Тестирование и отладка

## Планирование и проведение тестирования приложения.

Планирование тестирования приложения я начал уже на первых этапах строения приложения, так как уже на ранних этапах LOGGING ошибок дал мне преимущества. Само тестирование готового приложения я проводил на практике, в полевых условиях, так как основная деятельность нашей учебной группы проходит на компьютерах, то я решил что на время тестирования откажусь от других источников информации о расписании, это же и с подвигло к поиску и дебагу ошибок.

## Выявление и исправление ошибок и недочётов.

В основном недачёты найденные мной были связанны с плохо написанном JSON обработчиком данных, но с этим я справился. Одним из не исправленных багов осталось то что, после загрузки приложения, не стиль css не само расписание и задачи не отображаются, но после нажатия на любую кнопку приложение начинало работать. Как я выяснил чтобы это исправить нужно работать с много поточностью в java, но к сожалению у меня не получилось подключить много поточность к моему приложению.

## Оптимизация производительности и устранение узких мест.

Я считаю что я добился высокой оптимизации в своём приложении, но узкие места устранил не все, к примеру, в моё приложении есть кнопка под названием «darkStyle» как можно догадаться из названия он изменяет стиль css у приложения, и для первой смены стиля нужно дважды нажать на неё.

# Результаты и анализ .

## Оценка выполнения требований и достижения целей работы.

Я считаю что все поставленные мной задачи были выполнены и я достиг целей работы. Я разобрался в среде разработки графических приложений JavaFX научился большему функционалу языка Java в целом разработал готовое приложение и протестировал его на себе и своих одногруппниках.

## Анализ эффективности разработанного приложения

Проведя анализ эффективности своего приложения могу сказать что оно работает на строгую четвёрку из десяти, так как ещё много вещей в ней можно поправить и сделать более быстрым и практичным.

# Заключение

## Основные выводы работы.

В ходе выполнения данной работы было разработано JavaFX приложение, предназначенное для отслеживания расписания занятий и добавления заметок. Были достигнуты следующие результаты:

* Изучены возможности JavaFX для разработки графического интерфейса приложения.
* Проанализированы методы взаимодействия с внешними источниками данных для получения расписания занятий.
* Разработана функциональность добавления, редактирования и удаления заметок в приложении.
* Реализована интеграция с базой данных для хранения расписания занятий и заметок.
* Проведено тестирование приложения и оценена его производительность.

## Резюме выполненной работы.

В результате выполнения данной работы было создано удобное и интуитивно понятное JavaFX приложение, которое позволяет пользователям эффективно отслеживать своё расписание занятий и добавлять заметки для каждого дня. Приложение обладает функциональностью, необходимой для организации деятельности и управления временем. Благодаря интеграции с внешними источниками данных, пользователи могут получать актуальное расписание занятий.

## Рекомендации для дальнейшего развития и улучшения приложения.

На основе выполненной работы и полученных результатов, можно предложить следующие рекомендации для дальнейшего развития и улучшения приложения:

* Расширить функциональность приложения, добавив возможность установки напоминаний о предстоящих занятиях и важных событиях.
* Реализовать функцию синхронизации с календарными приложениями, чтобы пользователи могли иметь единое расписание на разных устройствах.
* Улучшить интерфейс приложения, сделав его более интуитивно понятным и привлекательным для пользователей.
* Провести дополнительное тестирование приложения с учётом разных сценариев использования и нагрузки, чтобы убедиться в его надёжности и производительности.
* Внедрить механизм обратной связи с пользователями, чтобы получать от них отзывы и предложения по улучшению приложения.

В целом, разработанное JavaFX приложение является полезным инструментом для эффективного управления временем и организации занятий. Оно имеет потенциал для дальнейшего развития и улучшения, чтобы удовлетворить растущие потребности пользователей и обеспечить им максимальную продуктивность.

# Список литературы

1. METANIT.COM - Сайт о программировании - https://metanit.com.
2. JavaFX Documentation (oracle.com) - https://www.oracle.com/java/technologies/javase/javafx-docs.html.
3. PostgreSQL: Documentation - https://www.postgresql.org/docs.
4. Java API for JSON Processing (oracle.com) - https://www.oracle.com/technical-resources/articles/java/json.html.
5. java.net (Java Platform SE 8 ) (oracle.com) - https://docs.oracle.com/javase/8/docs/api/java/net/package-summary.html.
6. JavaFX CSS Reference Guide (oracle.com) - https://docs.oracle.com/javase/8/javafx/api/javafx/scene/doc-files/cssref.html.
7. Logger (Java Platform SE 8 ) (oracle.com) - https://docs.oracle.com/javase/8/docs/api/java/util/logging/Logger.html

# Приложение

## Скриншоты интерфейса приложения



Рис 2. Окно верификации.

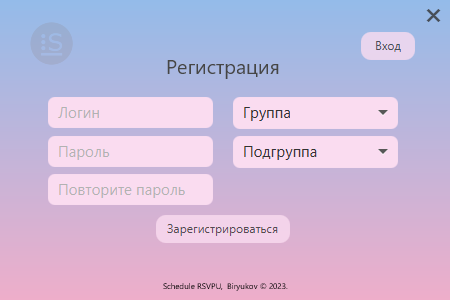


Рис 3. Окно регистрации.

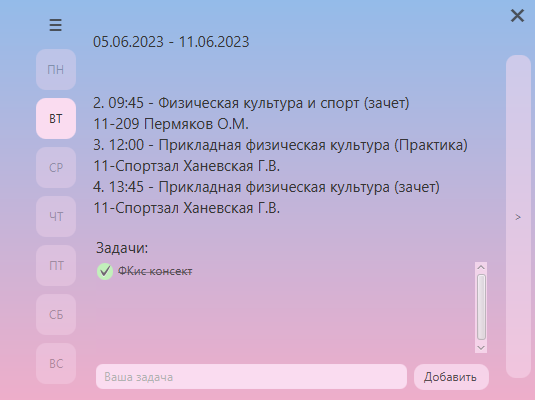


Рис 4. Главное окно приложения в светлой теме.

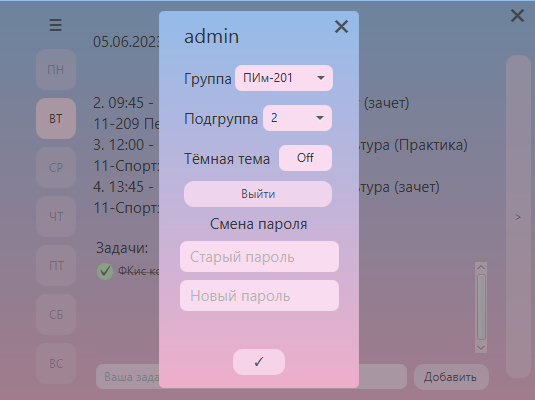


Рис 5. Окно настроек в сетлой теме.

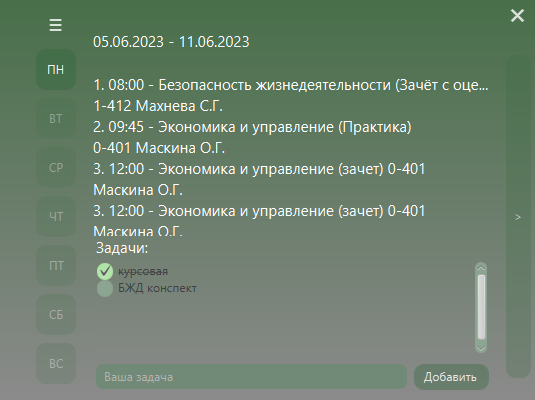


Рис 6. Главное окно приложения в тёмной теме.

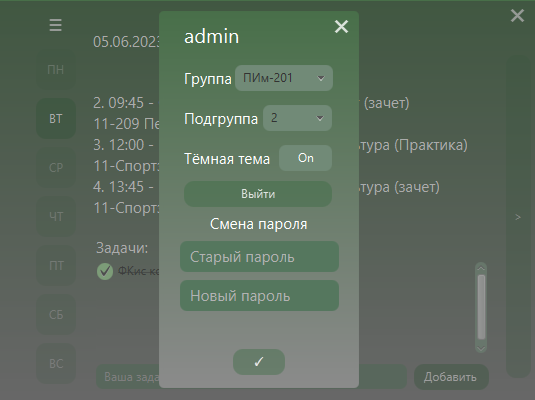


Рис 7. Окно настроек в тёмной теме.