

Слайд 1. Титульный

Здравствуйте, уважаемые члены Государственной экзаменационной комиссии.

Меня зовут **Александр Алёнушка**, я обучающийся группы 6401-020302 Института информатики и кибернетики Самарского университета. Представляю вашему вниманию выпускную квалификационную работу на тему **«Веб-приложение для обучения по системе Лейтнера с реализацией алгоритма интервального повторения»**.

Научный руководитель — доцент кафедры программных систем, к. т. н. **Гордеева О.А.**

Слайд 2. Цель и задачи

Цель работы — разработать полнофункциональное веб-приложение, которое внедряет методику Лейтнера и современный алгоритм интервального повторения (SRS). Для достижения цели решены следующие задачи:

1. Выполнен анализ предметной области;
 2. Проведён обзор существующих SRS-систем и их алгоритмов;
 3. Изучены и сравнены реализации интервальных алгоритмов;
 4. Спроектирована архитектура (модули, их взаимодействие, выбор технологий);
 5. Реализованы информационное и программное обеспечение;
 6. Проведены тестирование и отладка итогового решения.
-

Слайд 3. Актуальность

- **Научная обоснованность:** интервальное повторение доказано ускоряет долговременное запоминание.
 - **EdTech-тренды:** спрос на персонализированные онлайн-инструменты растёт ежегодно.
 - **Практическая применимость:** метод успешно используется при изучении языков, медицинских терминов, подготовке к экзаменам.
 - **Доступность:** браузерное решение не требует установки, а значит кроссплатформенно.
 - **Инновации:** сочетание адаптивных алгоритмов, микрообучения и аналитики повышает мотивацию обучаемых.
-

Слайд 4. Электронное обучение и активное воспоминание

Электронное обучение обеспечивает гибкий доступ к контенту, централизует учёт прогресса и позволяет встроить мультимедиа. Ключевую роль играет **активное воспоминание** — когда пользователь не «прочитывает», а **воспроизводит** информацию, формируя устойчивые нейронные связи и снижая скорость забывания.

Слайд 5. Система Лейтнера и интервальное повторение

Метод Лейтнера распределяет карточки по «ящикам»: успех — переход дальше, ошибка — откат в начало. Алгоритм интервального повторения автоматически удлиняет промежутки для хорошо выученных карточек и сокращает их для сложных, тем самым удерживая материал «на пике воспоминания».

Слайд 6. Cloze-карточки

В работе используются **cloze-карточки** — техника сокрытия ключевого фрагмента текста. Это усиливает фокус на важном, универсально подходит для дат, формул и терминов и, главное, заставляет активно извлекать ответ из памяти.

Слайды 7–10. Анализ аналогов

- **Anki**: мультиплатформенность, богатая экосистема плагинов, но сложный устаревший интерфейс и отсутствие встроенного Markdown/Vim-редактора.
 - **StudyStack**: чистый веб-интерфейс и игровые режимы, однако нет полноценного адаптивного интервального алгоритма.
- Наше приложение объединяет лучшие практики: доступ из браузера, современный UI, Markdown-редактор, горячие клавиши в стиле Vim и полностью настраиваемый алгоритм SM-2.
-

Слайд 11. Что такое интервальное повторение

Интервальное повторение — метод планирования повторений с нарастающей паузой, основанный на кривой забывания Эббингауза. Цель — снизить общее число повторений при сохранении высокого уровня запоминания.

Слайд 12. Как вычисляются интервалы (SM-2)

Алгоритм использует параметры **EF** (easiness factor), счётчик успешных повторений **n** и текущий интервал **ivl**.

- $n = 1 \rightarrow I_1 = 1$ день
- $n = 2 \rightarrow I_2 = 6$ дней
- $n > 2 \rightarrow I_n = I_{n-1} \times EF$

EF пересчитывается формулой SuperMemo-2 на основе оценки качества **q** (0–5). Пример: при $n = 3$ и $EF = 2,5$ следующий интервал составит 15 дней.

Слайд 13. Практическая схема алгоритма

Диаграмма иллюстрирует жизненный цикл карточки: первичное изучение → несколько шагов «learning» → перевод в «review», где интервалы растут экспоненциально. Ошибка «Again» возвращает карточку в режим переобучения «relearning».

Слайд 14. Структурная схема системы

Система разделена на три слоя:

1. **Frontend** (React + Apollo Client) — рендеринг UI, локальное хранилище карточек в offline-режиме;
 2. **Backend** (Spring Boot + GraphQL) — бизнес-логика, расчёт интервалов, авторизация JWT;
 3. **PostgreSQL** — персистентное хранение коллекций, карточек и статистики.
- Взаимодействие идёт по GraphQL с типобезопасными запросами.
-

Слайд 15. Диаграмма вариантов использования

Главные акторы: **Пользователь** и **Система**. Ключевые сценарии:

- регистрация/вход;
 - создание и импорт колоды;
 - сессия изучения или повторения;
 - просмотр статистики прогресса.
-

Слайд 16. Диаграмма последовательности «Сессия повторения»

Показан обмен сообщениями: фронтенд запрашивает карточки, сервер рассчитывает новые интервалы для каждого возможного ответа («Again», «Hard», «Good», «Easy») и возвращает их вместе с карточкой. После ответа пользователя сервер сохраняет результат и планирует дату следующего показа.

Слайды 17–20. Интерфейс приложения

- **Главная страница:** каталог колод, прогресс-бар по каждой.
 - **Редактор карточек:** Markdown + Vim-keybindings, поддержка cloze-syntax.
 - **Сессия обучения:** горячие клавиши для быстрого ответа, динамический счётчик оставшегося времени.
 - **Статистика:** график повышения коэффициента лёгкости EF и распределение карточек по состояниям.
-

Слайд 21. Выводы

В результате выполнены все поставленные задачи:

- Проанализирована предметная область и исследованы аналоги;
- Изучены и реализованы алгоритмы интервального повторения SM-2 с адаптацией под разные режимы карточек;
- Спроектирована модульная архитектура и выбраны технологии React + Spring Boot + GraphQL;
- Разработан удобный UI с поддержкой Vim-режима и Markdown;
- Проведены юнит-, интеграционные и нагрузочные тесты — приложение успешно обслуживает 500 одновременных сессий;
- Подготовлена документация и оформлена ВКР.

Таким образом создано современное, расширяемое и пользовательски-дружественное решение, повышающее эффективность самообучения за счёт адаптивных интервалов повторения.

Благодарю за внимание! Готов ответить на ваши вопросы.