Региональная конференция по научно-техническому творчеству школьников «Лабиринты науки»

Тема работы: «Разработка веб-приложения для проведения соревнований по программированию математических игр»

Выполнил:
Миронов Матвей Дмитриевич
МОУ СШ №43, 10 «ТЕХ» класс
Научный руководитель:
Бабошина Анна Викторовна
учитель информатики МОУ СШ №43

Ярославль
2023 г.

Оглавление

Введение **3**

Выбор фреймворка для разработки веб-приложения **4**

Выбор математических задач **5**

Разработка архитектуры веб-приложения **5**

Разработка интерфейса **9**

Заключение **10**

Список литературы **11**

Введение

В современном информационном обществе цифровые технологии становятся неотъемлемой частью нашей повседневной жизни и профессиональной деятельности, а развитие алгоритмического мышления становится все более важным. Алгоритмическое мышление представляет собой способность разрабатывать логические последовательности шагов для решения задач, анализа проблем и поиска оптимальных решений. Этот навык способствует креативному мышлению и развитию критического взгляда на задачи, что делает его одним из ключевых элементов современного образования. Навык алгоритмически мыслить помогает не только в сфере математики и информатики, но и в повседневных задачах, разработке инноваций и оптимизации процессов. Поэтому способность разрабатывать и оптимизировать алгоритмы становится важной частью образования. Именно в этом и заключается актуальность нашего проекта.

Одной из проблем, с которой сталкиваются ученики при развитии алгоритмического мышления, является отсутствие мотивации и интереса к данному навыку. В современном образовательном процессе помимо теоретических концепций и абстрактных примеров необходимо использовать интерактивные и практические методы обучения, игровые элементы, создавать соревновательную среду и возможности для практического применения полученных знаний.

Таким образом, **целью** данного проекта является разработка веб-приложения для проведения соревнований по программированию математических игр как интерактивного метода обучения алгоритмизации.

Задачи проекта:

1. Выбрать и ознакомиться с возможностями фреймворков для разработки веб-приложения.
2. Проанализировать и подобрать математические игры, которые будут доступны на платформе для реализации.

3. Определить функции веб-приложения и выбрать архитектуру.
4. Создать пользовательский интерфейс.

Практическая значимость работы состоит в разработке веб-приложения для развития алгоритмического мышления у учеников, что важно для их успешной адаптации к цифровой среде и повышения информационной грамотности. Также эта платформа создает соревновательную среду, которая стимулирует учеников к улучшению своих навыков и поиску наилучших решений. Данная платформа может быть использована в рамках образовательного процесса, как в университете, так и в школе. Помимо этого, на базе платформы могут быть реализованы соревнования в рамках тематической олимпиады.

Выбор фреймворка для разработки веб-приложения

Выбор языка программирования и фреймворка для создания веб-приложения является главным аспектом, который определит функциональность и производительность конечного продукта. На сегодняшний день есть множество языков программирования и фреймворков, каждый из которых имеют свои особенности, преимущества и недостатки. От Java и Python до JavaScript и Ruby - разнообразие выбора может ошеломлять.

Для разработки нашей платформы отлично подойдет язык программирования Java, поскольку он является одним из самых надежных, стабильных и эффективных. Благодаря строгой типизации и отличной поддержке многопоточности среди других языков, Java идеально подходит для создания веб-приложений, которым нужно обрабатывать множество запросов одновременно.

Определившись с языком программирования, нужно выбрать фреймворк. Для Java существует достаточно много различных вариантов, но мы остановимся на одном из самых популярных фреймворков – Spring [3].

Благодаря своей популярности, Spring имеет большое сообщество, которое создает огромное количество статей, видеоуроков и форумов, где всегда можно найти решение для различных проблем, связанных с этим фреймворков. Кроме того, Spring регулярно обновляется, улучшая безопасность, скорость и гибкость, что позволяет нам создать любое веб-приложение с использованием этого фреймворка.

Выбор математических задач

Для разработки веб-приложения нами были подобраны парные конечные математические игры. Цель каждой подобранной математической игры – поиск и анализ выигрышной стратегии. Каждая игра, представленная на платформе, дает возможность учащимся разработать и запрограммировать выигрышную стратегию на основе знаний, полученных или получаемых на занятиях по информатике и математике. Стоит отметить, что каждая игра предполагает возможность создания алгоритма, в соответствии с которым ходы будут производиться случайным образом, что дает возможность участия в создании стратегии игры учащемуся, как с высоким, так и с низким уровнем предметной подготовки. В свою очередь элемент игры призван мотивировать учащихся к победе, а именно – к актуализации знаний, поиску и изучению нового материала [4].

Разработка архитектуры веб-приложения

Перед началом создания веб-приложения надо четко определить, какие функции оно будет выполнять, какие компоненты системы будут присутствовать в приложении и какой архитектурный паттерн лучше выбрать.

Рассмотрим функции, которые должны присутствовать в нашем продукте. Веб-приложение должно иметь возможность загрузки алгоритмов пользователями для дальнейших соревнований с другими алгоритмами в

различных математических играх. Поэтому в первую очередь нам нужна возможность авторизации, которая позволит различать пользователей по ролям: учителя и ученики. Учителя могут создавать группу, организовывать игры, а также наблюдать за прогрессом своих учеников. Ученики же, в свою очередь, имеют возможность присоединяться к группам, загружать свои алгоритмы и участвовать в соревнованиях. Итак, ключевые функции нашего приложения - это система авторизации с разделением пользователей на учителей и учеников, возможность для учеников присоединяться к группам и загружать алгоритмы, а для учителей — создавать группы, организовывать игры и отслеживать прогресс участников.

Выбор архитектуры веб-приложения имеет очень важную роль в разработке [6]. Так как правильно выбранная архитектура позволяет обеспечить высокую производительность приложения, защиту данных от различных угроз и гибкость, позволяя добавлять новые функции с минимальными усилиями. На данный момент существует достаточно много различных архитектур, для нашей платформы мы выбрали один из самых популярных паттернов - MVC (Model-View-Controller) [2]. Как следует из названия данного шаблона, паттерн MVC имеет три слоя:

- Модель - представляет бизнес-уровень приложения
- Вид - определяет представление приложения
- Контроллер - управляет потоком приложения

В контексте языка программирования Java модель состоит из простых классов, вид отображает данные пользователю, а контроллер состоит из сервлетов. Таким образом, пользовательские запросы обрабатываются следующим образом (см. рис. 1):

1. Пользователь отправляет запрос контроллеру, который находится на сервере.
2. Контроллер обращается к модели, где в ответ на запрос извлекаются необходимые данные.

3. Затем контроллер передает полученные данные в вид.
4. Вид отправляется и отображается пользователю.

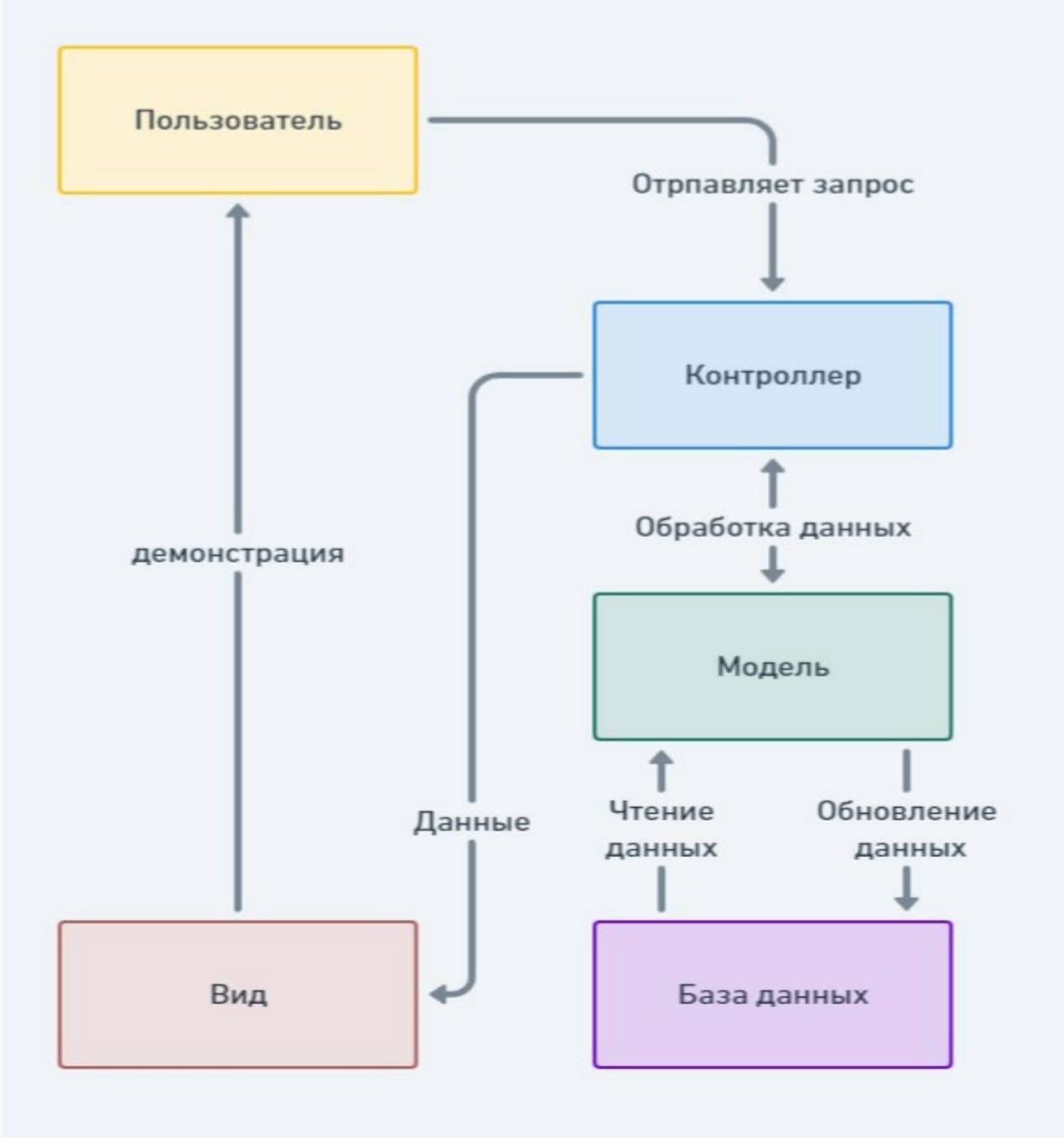


Рисунок 1. Схема работы архитектуры MVC

Разделение веб-приложения на три отдельных компонента имеет ряд преимуществ:

- Благодаря разделению приложения на компоненты, в него легче вносить изменения. Например, можно изменить внешний вид приложения, не затрагивая его основную бизнес-логику.
- Сокращение дублирования кода, так как компоненты MVC можно повторно использовать в различных частях приложения.
- Легкость тестирования, так как компоненты MVC можно тестировать независимо друг от друга.

Теперь мы можем рассмотреть, какие компоненты системы мы будем использовать и как, чтобы реализовать все функции приложения. В первую очередь реализуем систему авторизации. Для начала следует подключить базу данных MySQL к проекту [5], в ней мы будем хранить данные пользователей, групп и алгоритмы. После подключаем Spring Security и настраиваем SecurityConfig с целью добавления авторизации на платформу и контроля доступа к различным запросам в различии от роли пользователя (ученик / учитель) [8]. Реализовав авторизацию, перейдем к загрузке алгоритмов и организации игр. Чтобы сделать загрузку алгоритмов и сохранение их в базу данных, нам нужно создать и настроить контроллер. В этом контроллере мы будем получать от пользователя алгоритм, проверять его на работоспособность и соблюдение правил игры и сохранять в виде байтовой последовательности в базе данных. Для организации игр будет использоваться следующий алгоритм: после запроса от пользователя с ролью учителя о запуске игры в определенной группе, контроллер извлечет из базы данных информацию обо всех участниках и их алгоритмах в виде байтовых последовательностей. Игры будут проводиться согласно круговой системе – системе розыгрыша, при которой каждый участник турнира играет с каждым в ходе тура или раунда. Это позволяет максимально объективно и точно определить победителя. Однако следует отметить, что такой метод выявления победителя может занять значительное время. Для ускорения этого процесса мы будем использовать многопоточность. В Java

многопоточность позволяет программе выполнять несколько задач одновременно. В контексте нашего приложения это означает, что разные игры между участниками могут проводиться параллельно, что значительно повышает эффективность и скорость обработки [1]. Для реализации многопоточности мы будем использовать интерфейс ExecutorService, который позволяет создать пул потоков с фиксированным размером. Это означает, что одновременно сражаться будут не более определенного количества участников. В данный момент используется пул потоков размером в 10, но при увеличении количества участников или игр мы можем расширить его для оптимизации производительности. Важно найти баланс между количеством потоков и доступными системными ресурсами, чтобы избежать чрезмерной нагрузки на сервер. При использовании многопоточности также крайне важно учитывать синхронизацию, чтобы предотвратить состояние гонки (Race Condition) и другие проблемы совместного доступа к ресурсам [7]. Состояние гонки возникает, когда несколько потоков одновременно пытаются выполнить операции чтения или записи к общим ресурсам без должной синхронизации. В нашем приложении мы используем блоки synchronized для обеспечения безопасности при доступе к общим ресурсам. В каждом потоке мы будем запускать процесс Python-скрипта, соответствующего пользователю. Для этого мы используем класс ProcessBuilder, который позволяет создавать процесс из байтовой последовательности Python-скрипта, подавать входные данные в процесс алгоритма участника и получать выходные данные. После завершения всех игр результаты агрегируются, сортируются и передаются в модель для отображения пользователю.

Разработка интерфейса

В разработке веб-приложения важно также особо уделить внимание созданию функционального и эффективного пользовательского интерфейса.

Ключевым инструментом в нашей работе над интерфейсом является Thymeleaf - современный серверный Java-шаблонизатор. Thymeleaf позволяет нам легко отображать динамические данные. Использование этого шаблонизатора позволяет нам нам создавать более интерактивные, эффективные и пользовательски ориентированные интерфейсы, что значительно улучшает взаимодействие пользователя с нашим приложением.

Заключение

В ходе проделанной работы после выбора фреймворка и подбора математических игр, было разработано веб-приложение на фреймворке Spring с использованием архитектуры MVC. Была подключена база данных MySQL для организации хранения данных о пользователях и алгоритмах. Также был разработан функциональноый и эффективный пользовательский интерфейс, ключевым инструментом при разработке является современный серверный Java-шаблонизатор – Thymeleaf. Для ускорения работы платформы была использована многопоточность, позволяющая программе выполнять несколько задач одновременно.

Таким образом, поставленная нами цель достигнута, задачи решены. В дальнейшем планируется увеличение количества математических игр, наличие возможности решения задач на нескольких языках программирования (Python, Pascal, C++, C#, Java), а также увеличение серверных мощностей для проведения большего количества одновременных соревнований. Данное веб-приложение планируется использовать на базе Ярославского государственного педагогического университета им. К.Д. Ушинского в рамках курса «Развивающие игры» и в рамках внеурочной деятельности старших классов в МОУ СШ №43 им. А.С. Пушкина.

Список литературы

1. Javarush. Многопоточность в Java: [Электронный ресурс]. URL: https://clck.ru/36dFZa (дата обращения: 20.08.2023).
2. Skillbox. Что такое MVC: [Электронный ресурс]. URL: https://clck.ru/ue5R8 (Дата обращения: 13.08.2023).
3. Spring: [Электронный ресурс]. URL: https://spring.io/ (дата обращения: 13.08.2023).
4. Бабошина А.В., Корнилов П.А. О системе игр при обучении старшеклассников алгоритмам и математическим структурам // Информатика в школе №8 (171) октябрь 2021. Изд-во: «Образование и Информатика»
5. Дюбуа П. MySQL. Сборник рецептов. – Пер. с англ. – СПб: Символ Плюс, 2006. –1056 с., ил. ISBN 5-93286-070-7.
6. Лакамера Д. Архитектура встраиваемых систем. // Научный редактор: Романов А. Ю; Из-во: ДМК-Пресс; ISBN: 978-5-93700-206-8; С. 332.
7. Опивалов С.А. Методы работы с потоками в языке Java // International Journal of Humanities and Natural Sciences, vol. 4-3 (79), 2023.
8. Уоллс, К. Spring в действии / К. Уоллс. — Москва : ДМК Пресс, 2013. — 752 с. — ISBN 978-5-94074-568-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/39988 (дата обращения: 16.08.2023).