

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра дискретной математики и информационных технологий

**РАЗРАБОТКА ПРИЛОЖЕНИЯ ПО ФОРМИРОВАНИЮ  
УПРАЖНЕНИЙ ПО ФИТНЕСУ С ИСПОЛЬЗОВАНИЕМ  
ТЕХНОЛОГИИ SPRING FRAMEWORK**

**АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ**

студента 4 курса 421 группы  
направления 09.03.01 — Информатика и вычислительная техника  
факультета КНиИТ  
Калантарова Артема Наилевича

Научный руководитель  
доцент, к. ф.-м. н.

\_\_\_\_\_

О. В. Мещерякова

Заведующий кафедрой  
доцент, к. ф.-м. н.

\_\_\_\_\_

Л. Б. Тяпаев

Саратов 2024

## ВВЕДЕНИЕ

В последнее время создание веб-приложений является достаточно распространенной деятельностью и над удобством и простотой создания веб-приложения работает достаточно большое число компаний, которые постоянно обновляют различные библиотеки и фреймворки.

Библиотека в программировании представляет собой коллекцию уже написанных функций, классов и объектов, которые используются для решения различных задач [1].

Фреймворк — это набор инструментов и шаблонов, которые используются для создания продуктов или процессов в программировании и других областях. Предоставляет основу для разработки приложений, упрощает процесс разработки и повышает эффективность работы программистов [2].

Веб-приложение является прикладным программным обеспечением, логика которого распределена между сервером и клиентом, а обмен информацией происходит по сети. Клиентская часть реализует пользовательский интерфейс, а серверная — получает и обрабатывает запросы от клиента, выполняет вычисления, формирует веб-страницу и отправляет клиенту согласно протоколу HTTP.

HTTP (HyperText Transfer Protocol) — это протокол, который используется для передачи данных в Интернете. Позволяет получать различные ресурсы, такие как HTML-документы, и является основой для обмена данными в Интернете. HTTP работает по принципу клиент-серверного взаимодействия, где клиент (обычно веб-браузер) инициирует запросы к серверу для получения нужных данных [3].

Веб-приложение, созданное с применением современных методов разработки, представляет из себя информационный ресурс, благодаря которому можно осуществлять:

- Взаимодействие с пользователем и информационную поддержку.
- Рекламу, так как веб-приложение способно объединить в себе видеорекламу и баннеры [4].

В настоящее время забота о физической форме становится все более актуальной для людей. Не всегда легко самостоятельно составить эффективную программу тренировок. Разработанное приложение будет нести рекомендательный характер и не будет учитывать индивидуальные особенности

пользователя (за профессиональной консультацией следует обращаться к соответствующим специалистам), однако послужит хорошим ориентиром для достижения тех или иных спортивных целей.

В мире активного образа жизни и фитнеса много веб-приложений, однако большинство приложений коммерческой направленности и имеют большое количество рекламной информации, что вызывает дискомфорт у пользователей.

Целью работы является разработать веб-приложение, позволяющее формировать программы тренировок на основе индекса массы тела, а также предоставляющее пользователю возможность редактировать и составлять тренировочные планы самостоятельно.

Поставлены следующие задачи:

- Подготовить рабочую область для разработки с использованием сервера Tomcat. Добавить необходимые зависимости в проект (для механизма аутентификации и авторизации, валидации данных, тестов).
- Разработать контроллеры, репозитории, сервисы в рамках трехслойной архитектуры приложения.
- Использовать ORM для взаимодействия с базой данных.
- Подготовить программы тренировок с учетом мнения эксперта в соответствующей предметной области.
- Разработать функциональную составляющую формирования программ тренировок.
- Разработать адаптивный пользовательский интерфейс в соответствии с ГОСТ Р 52872-2019 «Интернет-ресурсы и другая информация, представленная в электронно-цифровой форме. Приложения для стационарных и мобильных устройств, иные пользовательские интерфейсы».
- Протестировать серверную часть приложения с использованием юнит-тестов.

## КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

**В первой главе** рассматриваются теоретические сведения об основных технологиях, которые использовались при разработке веб-приложения по фитнесу.

Java представляет собой язык программирования и платформу вычислений, которая была впервые выпущена компанией Sun Microsystems в 1995 году [5].

Java — это язык программирования, который часто используется для создания веб-приложений. Простой, эффективный и универсальный. Изначально Java была создана для встраиваемых сетевых приложений, которые работают на разных платформах. Java является объектно-ориентированным, переносимым и интерпретируемым языком.

Объектно-ориентированный язык программирования — это язык программирования, который позволяет разработчикам создавать программы, основанные на объектах, которые могут содержать данные и функции.

Java является кроссплатформенным языком программирования. Приложение, написанное на Java, будет работать на любом компьютере, если установлен интерпретатор Java. Кроме того, Java обладает набором функций безопасности, которые защищают компьютер пользователя от ошибок в коде и вредоносных программ. Например, пользователь может безопасно запустить Java-апплет, загруженный из Интернета, так как функции безопасности Java предотвращают доступ апплетов к жесткому диску и сетевым соединениям.

Апплет — это небольшая программа, которая предназначена для встраивания в другое приложение [6].

Кроссплатформенный язык программирования — это язык программирования, который позволяет создавать приложения, работающие на различных платформах, используя один и тот же код. Приложения могут быть запущены на нескольких платформах без необходимости внесения значительных изменений [7].

Java является интерпретируемым языком программирования. Исходный код Java компилируется в байт-код, который затем выполняется на виртуальной машине Java (JVM), которая работает как программный интерпретатор. Байт-код обеспечивает эффективность работы интерпретатора, а

также обеспечивает переносимость Java на любую JVM, которая правильно реализована, независимо от аппаратной или программной конфигурации компьютера. Большинство веб-браузеров содержат JVM для запуска Java-апплетов.

Байт-код — это промежуточный код, который создается при компиляции программы на высокоуровневом языке, например, Java. Байт-код состоит из инструкций, которые могут быть выполнены виртуальной машиной. Виртуальная машина интерпретирует байт-код и выполняет соответствующие операции. Байт-код используется для обеспечения переносимости программы между различными платформами, так как виртуальная машина может быть реализована на разных операционных системах и аппаратных платформах.

JVM (виртуальная машина Java) — это программное обеспечение, которое выполняет программу Java построчно [8].

Java может работать немного медленнее, чем C++, но обладает хорошими функциями безопасности и переносимостью. Виртуальная машина Java обеспечивает изоляцию между программой и компьютером, на котором выполняется. Синтаксис Java похож на C++, но присутствуют и отличия, например, Java не поддерживает перегрузку операторов. Java также является динамическим языком, что позволяет безопасно изменять программу во время выполнения, что важно для сетевых приложений. Кроме того, все основные типы данных Java предопределены и не зависят от платформы, в отличие от C++.

Динамический язык программирования — это язык программирования, который позволяет определять типы данных и выполнять синтаксический анализ и компиляцию во время выполнения программы. [9].

Перегрузка операторов — это способность использовать стандартные операторы языка для работы с различными типами данных, включая те, которые были созданы пользователем [10].

Java и C++ имеют различия в структуре программ. В Java все функции и операторы должны находиться внутри класса, в то время как в C++ можно определять функции и строки кода вне классов, как в стиле C. В Java нельзя использовать глобальные переменные и методы вне класса, в то время как в C++ это возможно. Такие ограничения в Java помогают обеспечить целостность и безопасность программ [11].

В контексте создания надежных и масштабируемых приложений разработчики часто обращаются к таким фреймворкам, как Spring. Spring предоставляет разработчикам инструменты для создания сложных систем, например многопользовательских корпоративных приложений с множеством функций для бизнеса. Spring позволяет быстро создавать приложения, которые умеют работать с базами данных и облаками, состоят из разных модулей, обмениваются данными с пользователями через интернет по защищенным каналам. Подобное можно реализовать в Java вручную, но Spring предоставляет разработчикам уже готовые инструменты, которые позволяют разрабатывать приложения эффективнее. Spring полностью бесплатный и имеет открытый исходный код. С помощью Spring создается серверное программное обеспечение для работы с современными десктопными, мобильными и веб-приложениями. Spring полезен как для простых сервисов, так и для крупных проектов из множества модулей и функций.

В процессе разработки Spring помогает:

- Проектировать приложения. Создавать фундамент, с которым в дальнейшем будут работать программисты.
- Реализовывать меньше рутинных функций. Spring содержит множество базовых инструментов, которые можно не разрабатывать. Например, быструю настройку полноценной интеграции с базой данных или передачу информации через Интернет.
- Объединять между собой разные модули приложения. Строить зависимости и подключать данные зависимости друг к другу без риска нарушить связи между компонентами.

При разработке Spring придерживаются трех основных принципов:

- Универсальность. Spring должен подходить практически для любых проектов — как фундамент, на основе которого можно строить приложения разных типов. Все привнесенные решения должны быть универсальными и применимыми для широкой аудитории.
- Легкость. Важно, чтобы фреймворк Spring минимально влиял на логику проекта и содержал функции, которые действительно нужны. Код, написанный на Spring, должен легко переноситься на другой фреймворк или инструмент.
- Интегрированность. Одна из главных задач Spring — обеспечивать свя-

зи между разными технологиями, которые используются в разработке. Поэтому для Spring критически важны инструменты, облегчающие взаимодействие компонентов внутри проекта.

Spring Framework — это большой набор разных модулей, каждый из которых нужен для работы над определенными приложениями или частями данных приложений. Рассмотрим основные модули фреймворка Spring:

- Инверсия управления. Данный модуль лежит в основе Spring и контролирует взаимодействие разных компонентов между собой. С помощью специальных аннотаций можно описать зависимости между компонентами, соединяя компоненты в единую архитектуру. В итоге компоненты работают самостоятельно, а связями между компонентами управляет специальный контейнер, что позволяет сохранять работоспособность системы.

Инверсия управления (Inversion of Control) — это абстрактный принцип, набор рекомендаций для написания слабо связанного кода, где каждый компонент системы должен быть как можно более изолированным от других, не полагаясь в своей работе на детали конкретной реализации других компонентов [12].

Аннотации — это пометки, с помощью которых происходит указание компилятору Java и средствам разработки, что делать с участками кода помимо исполнения программы. Аннотировать можно переменные, параметры, классы, пакеты. Можно разрабатывать аннотации самостоятельно или использовать встроенные [13].

- Модуль доступа к данным. Объединяет сразу несколько библиотек и отвечает за взаимодействие программ с базами данных. Данный модуль превращает объекты Java в записи базы данных, позволяет вносить изменения, организовывать быстрый и безопасный доступ к данным.
- Spring Cloud. Данный модуль позволяет интегрировать между собой разные элементы систем, соединенные по сети: компьютеры, серверы и другие. Внутри Spring Cloud существуют инструменты для маршрутизации запросов, балансировки нагрузки на отдельные сервисы, обмена сообщениями.
- Модуль аспектно-ориентированного программирования. С помощью данного модуля программу можно сделать более выразительной за счет

разделения на отдельно функционирующие части с разными зонами ответственности. Аспектно-ориентированное программирование позволяет неявно работать с разными функциями системы так, что функции не могут мешать работе друг друга, что улучшает архитектуру приложения и позволяет не отвлекаться от бизнес-задач.

- Модуль MVC (модель, представление, контроллер). Данный модуль реализует популярную схему веб-приложений — разделение на три части. Первая часть отвечает за данные, вторая — за отображение интерфейсов, третья — за изменение данных в ответ на действия пользователя.
- Модуль авторизации. Данный модуль позволяет настраивать авторизацию и аутентификацию в системе, что обеспечивает разделение прав пользователей и защиту паролей.
- Модуль бизнес-приложений. Представлен в виде фреймворка Spring Roo, который позволяет быстро создавать структуру крупных приложений и универсальные шаблонные решения.
- Модуль интеграции — Spring Integration предназначен для интеграций корпоративных приложений в единую систему: обмениваться сообщениями, активировать функции и строить единую архитектуру.
- Spring Boot. Представляет собой отдельный модуль, который упрощает настройку фреймворка Spring и ускоряет запуск проектов. Может автоматически сконфигурировать приложение и создать веб-сервер для запуска приложения. Большинство современных приложений Spring создается с помощью Spring Boot [14].



**Во второй главе** описывается процесс разработки веб-приложения.

После анализа основных технологий приступили к разработке приложения для формирования тренировочных программ по фитнесу с возможностью последующего экспорта в различных форматах. Была реализована функция конструктора для самостоятельного составления тренировочных планов, а также добавлена возможность определения подходящего вида спорта на основе результатов теста. Первоначально на официальном сайте Spring был сгенерирован архив с базовыми настройками приложения. В процессе разработки корректировались метаданные и настройки проекта. Итоговый архив был распакован и открыт в среде разработки IntelliJ IDEA, которая обеспечивает удобство использования, интеграцию с различными инструментами, автоматическую проверку кода и поддержку множества плагинов.

После открытия проекта в IntelliJ IDEA были настроены базовые параметры, такие как версии используемых технологий и конфигурация приложения через файл `application.properties`. Настройки включают установку порта для веб-сервера, управление схемой базы данных, а также параметры подключения к базе данных PostgreSQL. Данные настройки обеспечивают корректное взаимодействие приложения с базой данных и сервером.

Разработка велась с использованием трехслойной архитектуры, включающей слой представления, бизнес-логику и слой данных. На платформе Spring Boot данные компоненты реализуются через контроллеры, сервисы и репозитории соответственно. Важным аспектом разработки является внедрение зависимостей (Dependency Injection), что упрощает тестирование, поддержку и расширение приложений.

Безопасность данных является важным аспектом при разработке веб-приложений. Для защиты от атак SQL-Injection в приложении используются технологии Hibernate и Spring Data JPA. Hibernate преобразует данные между объектами приложения и таблицами базы данных, генерируя SQL-запросы на основе сущностей и критериев. Spring Data JPA расширяет возможности JPA и обеспечивает автоматическую генерацию безопасных SQL-запросов через репозитории, что предотвращает возможные атаки SQL-Injection.

Пользователь предоставляет информацию о своих физических параметрах, таких как рост, вес, возраст и пол, для вычисления индекса массы тела (BMI) по следующей формуле [15]:

$$BMI = \frac{m}{h^2}$$

В данной формуле  $m$  – масса тела в килограммах, а  $h$  – рост в метрах. На основе вычисленного  $BMI$  и возрастной категории пользователя определяется тип физической формы и формируется тренировочный план рекомендательного характера.

## ЗАКЛЮЧЕНИЕ

В настоящей работе было разработано веб-приложение по формированию и управлению тренировочными планами. Были использованы следующие основные технологии:

- Язык программирования Java.
- Spring — фреймворк для создания Java-приложений.
- Hibernate — фреймворк для работы с реляционными базами данных.
- Thymeleaf — шаблонизатор для создания веб-интерфейсов.
- PostgreSQL — реляционная система управления базами данных.
- Язык программирования JavaScript.
- HTML — язык гипертекстовой разметки.
- CSS — язык стилей, который определяет внешний вид элементов HTML.

Рассмотренные технологии были выбраны с целью сокращения времени разработки веб-приложений, предотвращения дублирования кода и обеспечения простоты поддержки, модернизации приложения.

На этапах выполнения были достигнуты следующие результаты:

- Создана рабочая среда для разработки с использованием сервера Tomcat и добавлены необходимые зависимости для аутентификации, авторизации, валидации данных и тестирования.
- Реализованы контроллеры, репозитории и сервисы в рамках трехслойной архитектуры.
- Настроен ORM для эффективного взаимодействия с базой данных.
- Разработаны тренировочные программы при содействии эксперта.
- Внедрены функции для создания и взаимодействия с тренировочными планами.
- Создан адаптивный пользовательский интерфейс, соответствующий требованиям ГОСТ Р 52872-2019.

Разработанная система включает веб-интерфейс для взаимодействия с серверной частью приложения и базой данных. Пользователи могут регистрироваться, проходить тест и получать тренировочные программы с возможностью экспорта в формате .json и .xlsx. Внедрен конструктор тренировочных планов с контролем ввода обязательных данных.

## Основные источники информации:

- 1 Библиотеки в программировании: для чего нужны и какими бывают: [Электронный ресурс] URL: [https://skillbox.ru/media/code/biblioteki\\_v\\_prog\\_ammirovanii/?ysclid=lnk6vltcok672524162](https://skillbox.ru/media/code/biblioteki_v_prog_ammirovanii/?ysclid=lnk6vltcok672524162) (дата обращения: 10.10.2023)
- 2 Что такое фреймворк: [Электронный ресурс] URL: <https://netology.ru/blog/06-2023-framework> (дата обращения: 11.10.2023)
- 3 Обзор протокола HTTP: [Электронный ресурс] URL: <https://developer.mozilla.org/ru/docs/Web/HTTP/Overview> (дата обращения: 11.10.2023)
- 4 Разработка веб-приложения для инвентаризации оборудования корпоративной сети КФУ на платформе ActiveMap: [Электронный ресурс] URL: [https://kpfu.ru/student\\_diplom/10.160.178.20\\_3594393\\_F\\_Diplom.pdf](https://kpfu.ru/student_diplom/10.160.178.20_3594393_F_Diplom.pdf) (дата обращения: 12.10.2023)
- 5 Общие вопросы: [Электронный ресурс] URL: [https://www.java.com/ru/download/help/whatis\\_java.html](https://www.java.com/ru/download/help/whatis_java.html) (дата обращения: 12.10.2023)
- 6 Class Applet: [Электронный ресурс] URL: <https://docs.oracle.com/en/java/javase/20/docs/api/java.desktop/java/applet/Applet.html> (дата обращения: 12.10.2023)
- 7 What Are Cross Platform Programming Languages: [Электронный ресурс] URL: <https://blogs.embarcadero.com/what-are-cross-platform-programming-languages/> (дата обращения: 12.10.2023)
- 8 Что такое среда выполнения для Java: [Электронный ресурс] URL: <https://aws.amazon.com/ru/what-is/java-runtime-environment/> (дата обращения: 12.10.2023)
- 9 Динамический язык программирования: [Электронный ресурс] URL: [https://developer.mozilla.org/ru/docs/orphaned/Glossary/Dynamic\\_programming\\_language](https://developer.mozilla.org/ru/docs/orphaned/Glossary/Dynamic_programming_language) (дата обращения: 13.10.2023)
- 10 Перегрузка в C++. Часть II. Перегрузка операторов: [Электронный ресурс] URL: <https://habr.com/ru/articles/489666/> (дата обращения: 13.10.2023)
- 11 Java Programming Language: [Электронный ресурс] URL: <https://www.sciencedirect.com/topics/computer-science/java-programming-language> (дата обращения: 13.10.2023)
- 12 IoC, DI, IoC-контейнер — Просто о простом: [Электронный ресурс] URL:

- <https://habr.com/ru/articles/131993/> (дата обращения: 25.04.2024)
- 13 Аннотации в Java. Не путать с комментариями: [Электронный ресурс]  
URL: [https://gb.ru/posts/java\\_annotatations](https://gb.ru/posts/java_annotatations) (дата обращения: 25.04.2024)
- 14 В Java всегда весна: почему фреймворк Spring считается самым популярным у Java-разработчиков: [Электронный ресурс] URL: <https://practicum.yandex.ru/blog/framework-spring-java/> (дата обращения: 25.04.2024)
- 15 Индекс массы тела: [Электронный ресурс] URL: <https://cmphmao.ru/node/235> (дата обращения: 01.06.2024)