2 ОБЗОР ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ

На сегодняшний день мир электронной коммерции и информационных технологий выдвигает все новые требования, касающиеся корпоративных веб-приложений. Они должны создаваться, и внедрятся за меньшие деньги, с большей скоростью и меньшими затратами ресурсов. Такие приложения обладают рядом преимуществ, касающиеся скорости, защищенности и надежности, обеспечиваемые серверными технологиями.

В основе работы любого веб-приложения лежат понятия «клиент» и «сервер». Клиент - это та часть приложения, которая отображает пользователю интерфейс, а сервер - это та часть приложения, которая готовит информацию для отображения. Существует множество подходов к созданию веб-приложений. Одним из таких является метод Representational state transfer (REST) . В общем случае REST является очень простым интерфейсом управления информацией без использования каких-то дополнительных внутренних прослоек. Каждая единица информации однозначно определяется глобальным идентификатором, таким как url-адрес. Каждый url-адрес в свою очередь имеет строго заданный формат [3].

При разработке **заданной программной системы(??)** будем придерживаться принципов методаREST. Также при разработке **программной системы(??)** будем использовать следующие фреймворки и программные средства:

* веб фреймворк Spring MVC;
* фреймворк Spring Security;
* систему управления базами данных MySQL;
* ORM библиотеку Hibernate;
* платформа Java/Java Enterprise Edition (Java EE);
* фреймворк для автоматизированной сборки проектов Apache Maven.

2.1 Веб фреймворк Spring MVC

Spring Framework – это легковесный открытый фреймворк, упрощающий разработку корпоративных (Java EE) приложений и нацеленный на использование простых JavaBean объектов. JavaBean – это обычный Java класс, который ни от кого не наследуется.

Spring Framework предоставляет большую свободу Java-разработчикам в проектировании, кроме того, он предоставляет хорошо документированные и лёгкие в использовании средства решения проблем, возникающих при создании приложений корпоративного масштаба[4].

Между тем, особенности ядра Spring Framework применимы в любом Java-приложении, и существует множество расширений и усовершенствований для построения веб-приложений на Java Enterprise платформе. Этот фреймворк предлагает последовательную модель и делает её применимой к большинству типов приложений, которые уже созданы на основе платформы Java. Считается, что Spring Framework реализует модель разработки, основанную на лучших стандартах индустрии, и делает её доступной во многих областях Java.

Spring Framework может быть рассмотрен как коллекция меньших фреймворков или фреймворков во фреймворке. Одним из них является фреймворк MVC, фреймворк который ориентирован на запросы. Аббревиатура MVC в названии Spring MVC расшифровывается как Model-view-controller (Модель – представление – контроллер).  MVC — схема использования нескольких шаблонов проектирования, с помощью которых модель данных приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента так, что модификация одного из компонентов оказывает минимальное воздействие на остальные[5].

Литература по данному **фреймворку(????????)**:

Крейг Воллс «Spring в действии» практически является гидом по Spring Framework. Книга охватывает все функции, инструменты и методы, включая Spring MVC, REST, Spring Security и многое другое. В книги доступно описано множество практических примеров. Она адресована пользователям различного уровня подготовки[4].

2.2 Фреймворк Spring Security

Безопасность имеет решающее значение для всех веб-приложений. Spring Security –  это [Java](https://ru.wikipedia.org/wiki/Java" \o "Java)/[Java EE](https://ru.wikipedia.org/wiki/Java_EE) [фреймворк](https://ru.wikipedia.org/wiki/%D0%A4%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA" \o "Фреймворк), предоставляющий механизмы построения систем аутентификации и авторизации, а также другие возможности обеспечения безопасности для промышленных приложений, созданных с помощью [Spring Framework](https://ru.wikipedia.org/wiki/Spring_Framework). Проект был начат Беном Алексом (Ben Alex) в конце 2003 года под именем «Acegi Security», первый релиз вышел в 2004 году. Впоследствии проект был поглощён Spring'ом и стал его официальным дочерним проектом. Впервые публично представлен под новым именем Spring Security 2.0.0 в апреле 2008 года[6].

Литература по данному **фреймворку(???????????)**:

Книга Питера Мулариена «Spring Security 3» является идеальным пособием для Java разработчика в сфере защиты веб - приложений. В первой части книги производится обзор концепций и методов защиты, а также настройка и конфигурации фрейморка. Вторая часть книги посвящена обзору распространенных сценариев интеграции, с которыми разработчики сталкиваются ежедневно, и решению сложных вопросов в мире информационной безопасности.

Данная книга предназначена для Java разработчиков, которые занимаются созданием веб-проектов и веб-приложений. Менее опытные пользователи также смогут использовать большую часть изложенного материала[7].

2.3 Система управления базами данных MySQL

База данных - набор сведений, хранящихся некоторым упорядоченным способом. Иными словами, база данных - это хранилище данных.

Система управления базами данных (СУБД) - это совокупность языковых и программных средств, которая осуществляет доступ к данным, позволяет их создавать, менять и удалять, обеспечивает безопасность данных и т.д. В общем СУБД - это система, позволяющая создавать базы данных и манипулировать сведениями из них[8].

MySQL — [свободная](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D0%9F%D0%9E) [реляционная система управления базами данных](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%A1%D0%A3%D0%91%D0%94). MySQL возникла как попытка применить [mSQL](https://ru.wikipedia.org/wiki/MSQL) к собственным разработкам компании: таблицам, для которых использовались [ISAM](https://ru.wikipedia.org/wiki/ISAM) — подпрограммы низкого уровня. В результате был выработан новый [SQL](https://ru.wikipedia.org/wiki/SQL)-интерфейс, но [API](https://ru.wikipedia.org/wiki/API)-интерфейс остался в наследство от mSQL.

MySQL является решением для малых и средних приложений. Входит в состав многих северов, таких как : WAMP, LAMP, Денвер и других. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы. Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа [MyISAM](https://ru.wikipedia.org/wiki/MyISAM), поддерживающие полнотекстовый поиск, так и таблицы [InnoDB](https://ru.wikipedia.org/wiki/InnoDB" \o "InnoDB), поддерживающие транзакции на уровне отдельных записей[10].

**Литература на данную *тематику*??????(или обозвать как то по другому)**

Книга Виктора Крёнке «Теория и практика построения баз данных. 8 – е изд.», написана в форме учебного пособия для студентов, специализирующихся в области информационных технологий. В ней освещается широкий круг теоретический и практический вопросов, связанных с разработкой и использованием баз данных. Книга отличается продуманностью структуры, живым и доступным языком изложения, а также большим количеством примеров, моделирующих типичные ситуации из практики делового мира[8].

Книга Линн Бейли «Изучаем SQL» посвящена языку запросов SQL и управлению базами данных. Материал в книге излагается постепенно, начиная с описания базовых запросов и заканчивая сложными манипуляциями с помощью объединений, подзапросов и транзакций. Книга является отличным практическим пособием для тех, кто пытается разобраться в организации и управлении базами данных. Особенностью данного издания является уникальный способ подачи материала, выделяющий серию «Head First» издательства O’Reily[9] .

2.4 ORM библиотека Hibernate

Hibernate — библиотека для языка программирования Java, предназначенная для решения задач объектно-реляционного отображения. Данная библиотека предоставляет лёгкий в использовании фреймворк для отображения объектно-ориентированной модели данных в традиционные реляционные базы данных.

Целью Hibernate является освобождение разработчика от значительного объёма сравнительно низкоуровневого программирования по обеспечению хранения объектов в реляционной базе данных. Разработчик может использовать Hibernate как в процессе проектирования системы классов и таблиц «с нуля», так и для работы с уже существующей базой данных.

Hibernate не только решает задачу связи классов Java с таблицами базы данных (и типов данных Java с типами данных SQL), но и также предоставляет средства для автоматической генерации и обновления набора таблиц, построения запросов и обработки полученных данных и может значительно уменьшить время разработки, которое обычно тратится на ручное написание SQL- и JDBC-кода. Hibernate автоматизирует генерацию SQL-запросов и освобождает разработчика от ручной обработки результирующего набора данных и преобразования объектов, максимально облегчая перенос приложения на любые базы данных SQL, используя при этом аннотации.

Java, так и в программах Java EE, выполняемых на сервере[12].

**Литература по данной библиотеке(????):**

Книга Эммануэля Бернара и Джона Гриффина «Hibernate Search in Action.1st ed» является практическим руководством для Java разработчиков, содержащая множество примеров реализации Hibernate Core. В книге пошагово разбираются основные принципы и методы использования библиотеки Hibernate[11].

Также существует документация от разработчиков, «Hibernate Reference Documentation», которая содержит полное описание всех методов данной библиотеки и некоторые практические примеры их реализации.

2.5 Платформа Java/Java EE

Java – один из самых важных и популярных компьютерных языков в мире. Java представляет собой язык программирования и платформу вычислений, которая была впервые выпущена Sun Microsystems в 1995 г. Существует множество приложений и веб-сайтов, которые не работают при отсутствии установленной Java, и с каждым днем число таких веб-сайтов и приложений увеличивается. Java отличается быстротой, высоким уровнем защиты и надежностью. От портативных компьютеров до центров данных, от игровых консолей до суперкомпьютеров, используемых для научных разработок, от сотовых телефонов до сети Интернет[14].

Внутри Java существуют несколько основных семейств технологий: Java SE, Java EE, Java ME, Java FX.

Java Platform, Enterprise Edition, сокращенно Java EE — набор спецификаций и соответствующей документации для языка [Java](https://ru.wikipedia.org/wiki/Java" \o "Java), описывающей архитектуру серверной платформы для задач средних и крупных предприятий.

Спецификации детализированы настолько, чтобы обеспечить переносимость программ с одной реализации платформы на другую. Основная цель спецификаций – обеспечить [масштабируемость](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%81%D1%88%D1%82%D0%B0%D0%B1%D0%B8%D1%80%D1%83%D0%B5%D0%BC%D0%BE%D1%81%D1%82%D1%8C)  приложений и целостность данных во время работы системы. Java EE во многом ориентирована на использование её через веб как в [интернете](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82), так и в [локальных сетях](https://ru.wikipedia.org/wiki/%D0%9B%D0%92%D0%A1).  Данные спецификации реализуются различными контейнерами. Контейнерами называются средства среды времени выполнения Java EE, предоставляющие размещенным на них компонентам определенные службы, например управление жизненным циклом разработки, внедрение зависимости, параллельный доступ и т. д. Такие компоненты используют точно определенные контракты для сообщения с инфраструктурой Java EE и с другими компонентами. Перед развертыванием они должны упаковываться стандартным способом (повторяя структуру определенного каталога, который может быть сжат в архивный файл). Java EE представляет собой расширенный набор функций платформы Java SE, что означает, что API-интерфейсы Java SE могут использоваться любыми компонентами Java EE.

Java EE является промышленной технологией и в основном используется в высокопроизводительных проектах, в которых необходима надежность, масштабируемость, гибкость[13].

**Литература по данной тематике(????):**

Книга Герберта Шилдта «Java. Полное руководство,8 – е изд.» представляет собой всеобъемлющее руководство по языку Java, описывающее его синтаксис, ключевые слова и фундаментальные принципы программирования. Также рассмотрена значительная часть библиотеки Java API. Книга разделена на четыре части, каждая из которых посвящена отдельному аспекту среды программирования.

Часть I представляет собой подробный учебник по языку Java, который раскрывает базовые понятия, описывается наследование, пакеты, интерфейсы, обработка исключений.

В части II описаны основные аспекты стандартной библиотеки интерфейса прикладного программирования Java. В ней раскрыты такие темы, как строки, ввод-вывод, инфраструктура коллекций, элементы управления графического интерфейса пользователя.

В части III рассмотрены три важные технологии Java: Java Beans Swing и сервлеты. Часть IV содержит главы с примерами реального использования Java[14].

Книга Энтони Гонсалвеса «Изучаем Java EE 7» представляет собой логичное пошаговое руководство, в котором подробно описаны многие спецификации и эталонные реализации Java EE 7. Работа с ними продемонстрирована на практических примерах. Книга написана ведущим специалистом по обработке запросов на спецификацию Java EE. В ней содержится информация, изложенная с точки зрения эксперта по технологиям Java для предприятий. Данное издание предоставляет возможность познакомится с новейшей платформой Java EE, исследовать использование API EJB и JP, и многое другое[13].

2.6 Фреймворк для автоматизированной сборки проектов Apache Maven

Apache Maven – фреймворк для автоматизации сборки проектов, специфицированных на XML-языке Project Object Model (POM).Активно используется в больших проектах и помогает осуществить процессы компиляции, создания jar, создания дистрибутива программы, генерации документации.

Для платформы Java существуют два основных инструмента для сборки: Ant и Maven. В отличие от Apache Ant, Maven обеспечивает декларативную, а не императивную сборку проекта. То есть, в файлах проекта pom.xml, которые использует для своей работы Maven, содержится его декларативное описание, а не отдельные команды. Все задачи по обработке файлов проекта Maven выполняет через плагины.

Главное преимущество Maven — это управление зависимостями. Редко какие проекты пишутся без использования сторонних библиотек. Эти сторонние библиотеки зачастую тоже в свою очередь используют библиотеки разных версий. Maven позволяет управлять такими сложными зависимостями. Что позволяет разрешать конфликты версий и в случае необходимости легко переходить на новые версии библиотек [15].

**Литература по данной тематике(????):**

Одной из главных книг на **данную тематику** является книга «Maven: The Definitive Guide» от Sonatype Company. В книге дается подробное описание процесса сборки программного обеспечения, с использованием фреймворка Maven. В первой части издания демонстрируются возможности инструмента на примере разработке небольших приложений, от идеи до внедрения. Вторая часть книги содержит описание возможностей Maven и его интеграция с разными средами разработки, такими как Eclipse, IntelliJ, NetBeans[16].