**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ**

**Кафедра веб-технологий и компьютерного моделирования**

**Веб-приложение по бронированию ресторанов**Курсовая работа

Мялика Антона Сергеевича

студента 2 курса, 1-31 03 08-02 Математическое и программное обеспечение мобильных устройств

Шубенок Валентины Владимировны

студентки 2 курса, 1-31 03 08-02 Математическое и программное обеспечение мобильных устройств

Научный руководитель:  
Старший преподаватель кафедры веб-технологий Вельченко Сергей Александрович

Минск, 2020

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Факультет \_\_\_**ММФ**\_\_\_ Кафедра \_\_\_**ВТиКМ**\_\_\_

**«УТВЕРЖДАЮ»**

Зав. кафедрой \_В.М. Волков

«\_\_» \_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_г.

**ЗАДАНИЕ**

**НА ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ**

**Студентам** Мялику А.С. и Шубенок В.В

**курса** 2

**1. Тема курсовой работы** Создание веб приложения по бронированию ресторанов с использованием фрэймворка Spring

утверждена на заседании кафедры протокол № \_\_2\_\_\_\_ от «\_30\_\_» \_\_\_\_сентября\_\_\_\_ 2015\_ года

**2. Срок сдачи студентом законченной работы** \_\_\_\_\_\_\_\_\_\_23 мая 2020 года\_\_\_\_\_\_\_\_

**3. Библиографические описания источников, рекомендуемых студентам к ознакомлению при выполнении работы:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**4. Перечень вопросов подлежащих разработке или краткое содержание работы**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**5. Календарный график работы на весь период** (с указанием этапов работы и сроков их выполнения). **Даты консультаций и промежуточных отчетов.**

\_\_1.Чтение литературы и изучение материалов(8.10.2019 – 10.11.2019)

\_\_2.Изучение технологий создания проекта (11.11.2019 – 5.12.2019)

\_\_3.Постановка задач (6.12.2019 – 7.01.2020)

\_\_4.Решение задач (12.03.2019 – 20.04.2020)

\_\_5. Подготовка отчёта и презентации (21.04.2019 – 10.05.2020)

**6. Задание принял к исполнению** «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись студентов

**Руководитель** Вельченко С.А. / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оглавление

[Введение 5](#_Toc41136672)

[Глава 1. Теоретические сведения 7](#_Toc41136673)

[1.1 Назначение и возможности разработанного приложения 7](#_Toc41136674)

[1.2 Описание вариантов использования 8](#_Toc41136675)

[1.2.1 Предметная область 10](#_Toc41136680)

[1.2.2 Описание средств разработки приложения 11](#_Toc41136681)

[1.2.3 База данных 15](#_Toc41136682)

[Глава 2. Практическая часть 17](#_Toc41136683)

[2.1 Описание разработки практической части 17](#_Toc41136684)

[2.2 Постановка задачи 18](#_Toc41136685)

[2.3 Описание функционала 19](#_Toc41136686)

[2.3 Описание архитектуры информационной модели 21](#_Toc41136687)

[Глава 3. Организационная часть 23](#_Toc41136688)

[Заключение 24](#_Toc41136689)

[Список использованных источников 27](#_Toc41136690)

# Введение

На сегодняшний день сложно, почти невозможно вообразить себе жизнь, в которой нет современных гаджетов, технологий и электронных устройств. Они превратились в неотъемлемую часть жизни людей, они применяются в совершенно различных видах деятельности человечества, их можно найти в любой сфере жизни человека: в экономической, политической, социальной, духовной и так далее. Технический прогресс постоянно развивается, и с каждым днём мы становимся свидетелями всё новых и новых новинок и совершенствований в электронной технике, новых открытий в сфере информационных технологий, большим влиянием IT- и интернет-технологий в жизни человека. Иногда просто кажется немыслимым, что в прошлом человечество жило без доступа к электронным технологиям. В случае обращения к научной точке зрения, то IT представляет собой весь всесторонний накопленный опыт человеческой цивилизации, пригодный для использования на практике. Их используют в науке, бизнесе, учебе, работе, В том числе в промышленности и производстве материалов, медицине, архитектуре, моделировании и во множестве других сфер деятельности людей.

Мобильные телефоны, без сомнений, также имеют большое значение в нашем мире на сегодняшний день. Более чем у 86% населения нашей планеты есть мобильный телефон. Некоторые люди даже предпочитают больше времени проводить не с семьей или друзьями, а именно со своим телефоном. Даже когда человек спит, его телефон работает: может скачивать информацию для дальнейшего использования, обновляться и т.д. Телефон многим помогает и в решении повседневных задач.

С ростом значимости телефона в нашей жизни, значимость приложений также пропорционально растёт. Соответственно, перед многими владельцами интернет-ресурсов встаёт вопрос выбора между web- и мобильным приложением. Чтобы выбор был сделан, важно понимать, что из себя представляет каждое из них, оценить положительные и отрицательные стороны каждого варианта.

Web-приложение – это веб-сайт с конечное содержание страниц которого определяется по запросу пользователя. Плюсом является его кроссплатформенность: не важен вид операционной системы, установленной на устройстве.

Мобильное приложение – программное обеспечение, предназначенное для мобильных устройств (смартфонов, планшетов и т.д.) и адаптированное под определённую платформу (iOS, Android, Windows). В сравнении с web-ресурсами, его работа осуществляется без доступа к сети.

Без сомнения, достоинством веб-приложения является то, что они не требуют установки на устройство, то есть они не занимают ценную память устройства, чем не могут похвастаться мобильные приложения.

Несмотря на вышеуказанные недостатки, стоит также отметить, что в плане функционала, разработка мобильного приложения даёт большие возможности, чем клиент-серверная разработка. Тем не менее, затрат и времени на неё нужно больше.

Одним из преимуществ разработки веб-приложений также является тот факт, что обновлять их можно в любое время, при этом с этим не возникает никаких сложностей: для обновления веб-приложения, его необходимо обновить лишь на сервере, после чего каждый клиент будет работать с новой версией.

Также не следует забывать и то, что мобильное приложение почти в каждом случае служит дополнением к веб-составляющей: компания изначально разрабатывает своё веб-приложение, после чего переходит к разработке мобильного приложения.

Именно перечисленным выше причинам, как мне кажется, web-приложения заслужили широкую популярность и распространение, которые и привели к уровню развития данной отрасли, который мы наблюдаем сейчас. Именно поэтому подавляющее большинство людей сейчас пользуется всемирной паутиной - интернет.

# Глава 1. Теоретические сведения

## 1.1 Назначение и возможности разработанного приложения

Описание проекта: Система “JustBook” представляет из себя многопользовательское веб-приложение, с помощью которого пользователи могут забронировать онлайн столик в заведении, выбранном из каталога, указав время и дату бронирования, а также количество гостей/посетителей. Так же можно посмотреть список рекомендованных заведение и выбрать исходя из заданного ценового диапазона. И пользователи могут оценить заведение и оставить комментарий. В это же время владельцы заведений могут добавлять свои кафе либо рестораны в каталог.

Актуальность: На данный момент на территории Беларуси практически нет подобного реализованного проекта. Среди конкурентов максимум кого мы могли бы выделить, так это Relax.by – интернет-портал о развлечениях и городской жизни, а также каталог и сервис по бронированию заведений и услуг.

Несомненно, Relax.by достаточно популярен, за счет чего большое количество заведений хотят попасть в их базу, и соответственно у них имеется широкий выбор заведений, и они собраны в более разнообразные категории. Это конечно же можно отнести к плюсы данного портала. Но в тоже время у них отсутствует возможность совершить онлайн бронирование (на портале Relax.by, а также в их приложении указаны телефоны заведений, позвонив по которым вы можете забронировать столик). Следовательно, тратится больше времени на то, чтобы забронировать столик. Так же в их базе находятся не только рестораны, кафе и бары, но и другие заведение, например, салоны красоты и т.п., в то время как наш проект более узко направленный, что позволяет более качественно выполнять работу, и пользователям будет достаточно удобно использовать наш продукт.

## 1.2 Описание вариантов использования

## 

## 

## 

## 

## 1.2.1 Предметная область

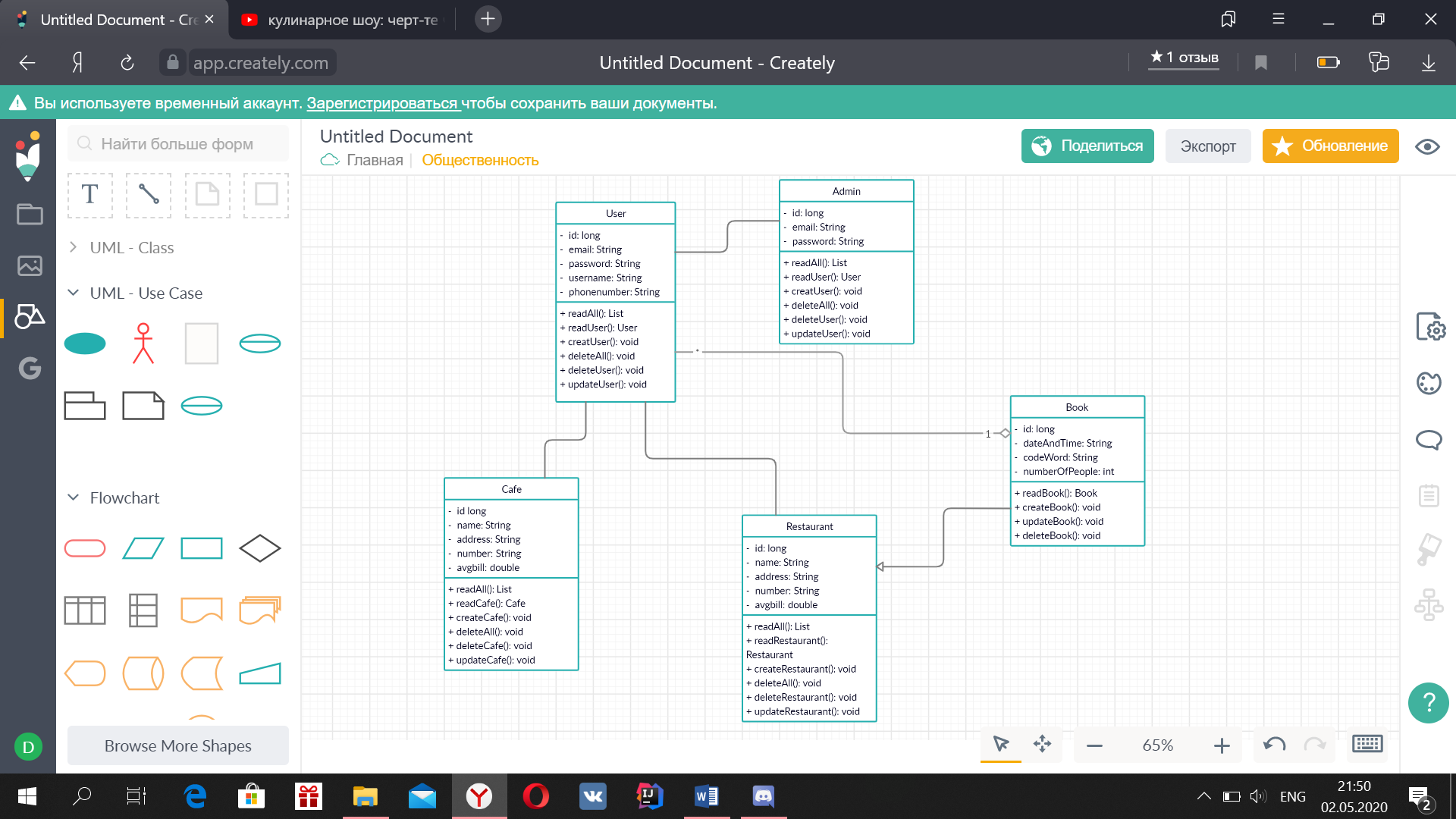


Диаграмма классов предметной области

## 

## 1.2.2 Описание средств разработки приложения

Объектно-ориентированный язык Java, разработанный в компании Sun Microsystems в 1995 году для оживления графики на стороне клиента с помощью апплетов, в настоящее время используется для создания переносимых на различные платформы и операционные системы программ. Язык Java нашел широкое применение в Интернет-приложениях, добавив на статические и клиентские веб-страницы динамическую графику, улучшив интерфейсы и реализовав вычислительные возможности.

Создание языка Java - это действительно один из самых значительных шагов вперед в области разработки сред программирования за последние 20 лет. Язык HTML (Hypertext Markup Language - язык разметки гипертекста) был необходим для статического размещения страниц во “Всемирной паутине” WWW (World Wide Web). Язык Java потребовался для качественного скачка в создании интерактивных продуктов для сети Internet.

Апплет – мини-программа, которая может исполняться только внутри браузера. Апплеты автоматически загружаются в составе веб-страниц. Когда апплет активизируется, он выполняет программу. Это одно из преимуществ апплета – он позволяет автоматически распространять программы для клиентов с севера именно тогда, когда пользователю понадобятся эти программы, и не раньше. Пользователь получает самую свежую версию клиентской программы, без всяких проблем и трудностей, связанных с переустановкой.

Если ваша программа исполняется в Интернете, то вы не знаете, на какой платформе ей предстоит работать. Приходится проявлять особую осторожность, чтобы избежать распространение некорректного кода. Здесь нужны межплатформенные и безопасные решения, наподобие Java или языка сценариев.

Самой сильной стороной Java считается серверное программирование. Чаще всего запрос сводится к простому требованию «отправьте мне этот файл». Более сложный запрос к серверу обычно связан с обращением к базе данных. В самом распространенном случае делается запрос на сложный поиск в базе данных, результаты которого сервер затем преобразует в HTML- страницу и посылает вам. А может быть, вам понадобится зарегистрироваться в базе данных при присоединении к какой-то группе, или оформить заказ, что потребует изменений в базе данных. Подобные запросы должны обрабатываться неким кодом на сервере; в целом это и называется серверным программированием. Для этого могут быть использованы веб-серверы, основанные на Java, позволяющие заниматься серверным программированием на Java с помощью так называемых сервлетов. Сервлеты и их детища, JSPs, составляют две основные причины для перехода компаний по разработке веб-содержимого на Java, в главном из-за того, что они решают проблемы несовместимости различных браузеров.

Наибольшее распространение получили сервлеты, обрабатывающие клиентские запросы по протоколу HTTP. Технология сервлетов является оболочкой протокола HTTP и поддерживает его как транспорт передачи данных от клиента к серверу и обратно.

Несмотря на появление множества успешных приложений, созданных на основе спецификации EJB, она никогда не применялась по прямому назначению: для упрощения разработки корпоративных приложений. Модель программирования EJB упрощает многие инфраструктурные аспекты разработки, такие как транзакции и безопасность. Но она принесла другие сложности, требуя создания дескрипторов развертывания и использования шаблонного кода. С течением времени многие разработчики разочаровались в EJB, что привело к спаду популярности данной технологии и поиска более простых путей.

Сегодня разработка Java-компонентов вернулась к своим истокам. Новые технологии программирования, включая аспектно-ориентированное программирование (AOP) и внедрение зависимостей (DI), дают JavaBeans дополнительные возможности, ранее заложенные в EJB. Эти технологии оснащают обычные Java-объекты (Plain-Old Java Objects, POJO) моделью декларативного программирования, напоминающей EJB, но без всей сложности спецификации EJB. Больше не надо создавать громоздкий компонент.

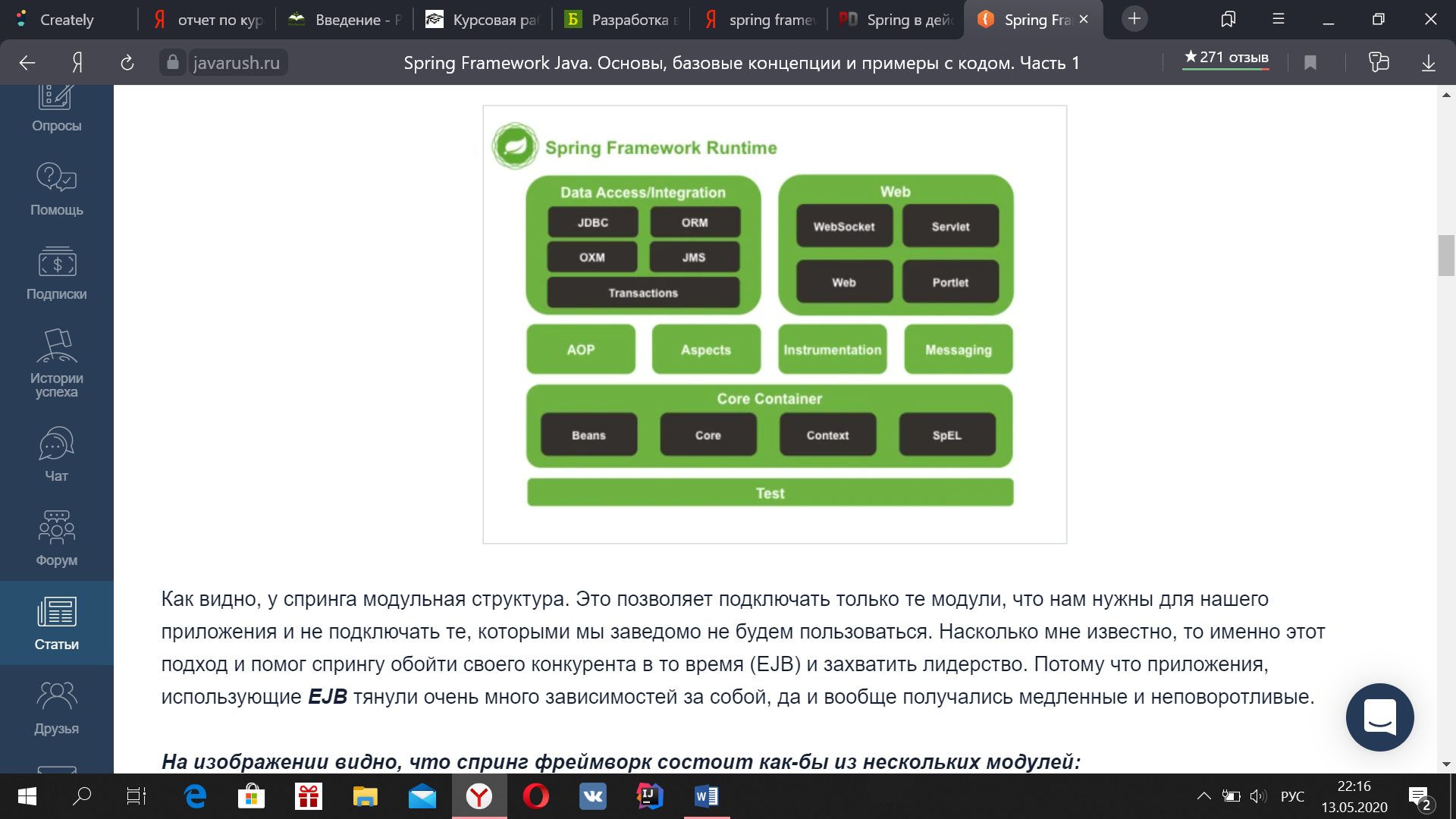
Фреймворк Spring Framework является передовым средством разработки приложений на основе POJO. Spring Framework, или просто Spring – один из самых популярных фреймворков для создания веб-приложений на Java. Фреймворк – это что-то похожее на библиотеку, но есть один момент. Грубо говоря, используя библиотеку, вы просто создаете объекты классов, которые в ней есть, вызываете нужные вам методы, и таким образом получаете нужный вам результат. Т.е., тут более императивный подход: вы четко указываете в своей программе в какой конкретный момент надо создать какой объект, в какой момент вызвать конкретный метод и т.д.. с фреймворками дела обстоят иначе. Вы просто пишите какие-то свои классы, прописываете там какую-то часть логики, а создает объекты ваших классов и вызывает методы за вас уже сам фреймворк.

В своем устремлении на сложности, связанные с разработкой на языке Java, фреймворк Spring использует четыре ключевые стратегии:

- легковесность благодаря применению простых Java-объектов (POJO);

- слабое связывание посредством внедрения зависимостей и ориентированности на интерфейсы;

- декларативное программирование через аспекты и общепринятые соглашения;  
- уменьшение объема типового кода через аспекты и шаблоны.



На рисунке выше основные модули **Spring**разбиты на несколько групп.

В рамках данного проекта Spring предлагает специальную абстракцию в виде интерфейса Repositoryи способен генерировать за вас не просто целые классы доступа к данным, а даже реализации некоторых методов по сигнатуре вашего интерфейса.

Web:

Spring MVC – модуль для создания обычных веб-приложений. Он, как следует из названия, полностью следует использованию паттерна MVC. Для реализации самого UI можно не ограничиваться одной Java и таких технологий как, например, JSF или ApacheVelocity, поддержка которых присутствует в Spring.

Spring Reactive Web – новый модуль Spring. Идеологическое продолжение проекта Spring Web Flux, с помощью которого вы можете максимально просто и красиво писать реактивные веб-приложения на Java.

AOP/Aspects:

Cредства поддержки использования аспектно-ориентированного программирования в своих приложениях. Реализация сквозной функциональность вроде логирования, транзакционности или безопасности может быть выполнена при помощи проекта Spring AOP. При этом Spring не запрещает, а даже помогает вам внедрить в свой проект другие, более мощные средства для поддержки аспектно-ориентированного программирования с применением хорошо зарекомендовавшей себя библиотеки AspectJ.

На самом деле Spring содержит в себе еще большое число различных модулей, среди которых Spring Messaging, Spring WS, Spring Batch, Spring Cloudи так далее.

Spring Boot помогает программисту максимально быстро запустить его проект, абстрагируясь от огромного количества деталей конфигурации в XML и настроек среды развертывания в виде контейнера. Spring Boot для этих целей предоставляет встроенный контейнер, который просто работает.

Модельная структура Spring позволяет подключать только те модули, что нам нужны для нашего приложения и не подключать те, которыми мы заведомо не будет пользоваться. Этот подход помог Spring обойти своего конкурента в то время (EJB) и захватить лидерство. Потому что приложения, которые использовали EJB тянули очень много зависимостей за собой, да и вообще получались медленными и неповоротливыми.

## 1.2.3 База данных

Oracle Database - это объектно-реляционная система поддерживающая некоторые технологии, реализующие объектно-ориентированный подход, то есть обеспечивающих управление создания и использования баз данных.

Ключевые возможности Oracle Database

• Real Application Cluster (RAC) обеспечивает работу одного экземпляра базы данных на нескольких узлах grid, позволяя управлять нагрузкой и гибко масштабировать систему в случае необходимости.

• Automatic Storage Management (ASM) позволяет автоматически распределять данные между имеющимися ресурсами систем хранения данных, что повышает отказоустойчивость системы и снижает общую стоимость владения (TCO).

• Производительность. Oracle Database позволяет автоматически управлять уровнями сервиса и тиражировать эталонные конфигурации в рамках всей сети.

• Простые средства разработки. Новый инструмент разработки приложений HTML DB позволит простым пользователям создавать эффективные приложения для работы с базами данных в короткие сроки.

• Самоуправление. Специальные механизмы Oracle Database позволяют самостоятельно перераспределять нагрузку на систему, оптимизировать и корректировать SQL-запросы, выявлять и прогно- зировать ошибки.

• Большие базы данных. Теперь максимальный размер экземпляра базы данных Oracle может достигать 8 экзабайт.

• Недорогие серверные системы. Oracle Database может использовать недорогие однопроцессорные компьютеры или модульные системы из “серверов-лезвий”.

• В новой версии базы данных реализована поддержка переносимых табличных пространств, система управления потоками данных Oracle Streams и модель распределенных SQL-запросов. Для переноса существующих баз данных в среду Grid в них не потребуется вносить изменений, что позволяет быстро начать использовать все преимущества Oracle Database.

# Глава 2. Практическая часть

## 2.1 Описание разработки практической части

Основной цель создания веб приложения по бронированию ресторанов - это разработка веб интерфейса который позволяет владельцам кафе и ресторанов рекламировать свои заведения с помощью каталога расположенного на сайте. А так же дать возможность беспрепятственно и без излишних звонов, сообщений и подтверждений увидеть количество свободных мест в заведении и в последствии забронировать их.

Основной целью практически каждого веб сервиса является приносить прибыль своему создателю, то есть владельцу( акционерам или другим лицам вложившимся в проект ). На данный момент не существует серьёзных аналогов которые могли бы что-то предложить в данной сфере,  
именно по этому основной доход будет идти от рекламы самих заведений в нашем веб-сервисе( в случае популярности )

Разработка и выпуск в массы данного продукта состоит из ряда важных этапов. От которого зависит возымеет успех данный сервис  
или нет. Среди всех этапов можно выделить основные пункты:  
- Анализ конкурентов на территории распространения и за её пределами( для понимания какие минусы и плюсы имеет готовый проект )  
- Поиск и разработка оптимального дизайна для данной сферы услуг. Не для кого не секрет что большинство заказов, продаж и рекламы  
не возможны без толкового и привлекающего внимания дизайна  
- Оптимизация приложения. Исходя из прочитанных фактов, если загрузка страницы длиться более 4 секунд то клиент покидает её и ищет альтернативу  
- Адаптация под мобильные устройства. По данным сервиса яндекс за 2016 год, около 23% запросов в браузере осуществляться с телефона  
- Определение функциональности сервиса, сопоставление с возможностью разработки и план реализации.  
- Помощь в указании какие действия должен совершить посетитель сервиса( При помощи дизайна, цветов, знаков и т.д. )

## 2.2 Постановка задачи

В наше время информационные технологии активно развиваются. Больше и больше людей начинают использовать всемирную сеть интернет для своих нужд. Поэтому начинает появляться больше программ, упрощающих поиск и систематизацию полученной информации, которые хранятся в базах данных.  
База данных - это организованный набор информации, обычно хранящихся и доступных в электронном виде из компьютерной системы. Там где базы данных являются более сложными они часто разрабатываются с использованием формальных методов проектирования и моделирования  
Использование баз данных для систематизации полученной информации значительно упрощается дальнейшее использование, мониторинг, хранение данных. Особенно это важно в настоящее время, когда все очень быстро меняется, например, каталог кафе и ресторанов в крупном городе.

В связи с этим было решено создать приложение, позволяющее просматривать заведения Минска, а также быстро и удобно бронировать их без лишних звонков и разговоров.

Т.е. веб-приложение, созданное в ходе выполнения курсовой работы должна представлять пользователю следующие возможности:

* 1. Удобный и доступный веб-интерфейс с основной информацией и необходимыми данными для успешного запуска данного приложения.
* 2. Предоставить актуальную на данный момент базу заведений
* 3. Занесение в базу данных информации введенной пользователем: Имя, персональный id, ссылка на бронь.
* 4. Предоставление пользователю возможности добавлять, удалить и исправлять внесенные им данные.
* 5. Предоставление пользователю упрощенного доступа к данным содержащимся в БД мониторинга.
* 6. Дать возможность прорекламировать своё заведение через веб-приложение возможному владельцу

## 2.3 Описание функционала

Цель работы: Спроектировать и разработать web-приложение для создания системы для людей, которые хотят забронировать столик в ресторане онлайн, не тратя время на звонки.

Описание пользовательской системы: Четыре основных класса пользователей – неавторизированный пользователь (Гость), авторизированный пользователь (Пользователь), авторизированный пользователь с правами администратора (Администратор) и авторизированный пользователь с расширенными правами пользователя (Владелец заведения)

*Гость* – человек, попавший на сервис с целью просмотреть, что из себя представляет наше приложение, какие возможности предоставляет, а также просмотреть наш каталог заведений.

*Пользователь* – человек, попавший на сервис с целью просмотреть имеющиеся заведения в каталоге, возможно, какие-то отзывы об интересующем его заведение, а также забронировать столик в ресторане, чтобы не тратить время и не делать это посредством телефонного звонка. Среднестатистический пользователь – человек 18-25 лет со средним-высоким уровнем достатка.

*Администратор* – пользователь, который использует сервис для курирования заведений и пользователей, т.е. обладает возможностями администрирования, например, удаляет или удаляет из базы заведение.

*Владелец заведения* – пользователь, который использует сервис в качестве главной или дополнительной рекламы, а также для снижения времени на оформления бронирования.

|  |  |
| --- | --- |
| Actor | Описание |
| Гость | Незарегистрированный пользователь - просматривать каталог ресторанов и кафе - читать отзывы - регистрация - авторизация |
| Пользователь | Авторизированный пользователь - все то, что может делать гость (кроме авторизации и регистрации) - оценить или оставить комментарий - изменить данные аккаунта - забронировать ресторан или кафе - отменить бронь заведения |
| Администратор | Авторизированный пользователь с правами администратора  - управление веб-приложением управление пользователями  управление сервисами управление контентом (ресторанами, кафе) - Управление бронированием просмотр удаление (в случае некорректных данных) |
| Владелец заведения | Авторизированный пользователь с расширенными правами пользователя - изменение данных аккаунта - просмотр каталога ресторанов и кафе - добавление ресторана или кафе в базу - изменение контента собственного заведения - просмотр оценки или отзывов |

## 2.3 Описание архитектуры информационной модели

База данных приложения предназначена для хранения используемых в приложении сущностей и быстрого доступа к ним. Эти сущности – Пользователь (User), Администратор (Admin), Кафе (Cafe), Бронирование (Book) и Ресторан (Restaurant).

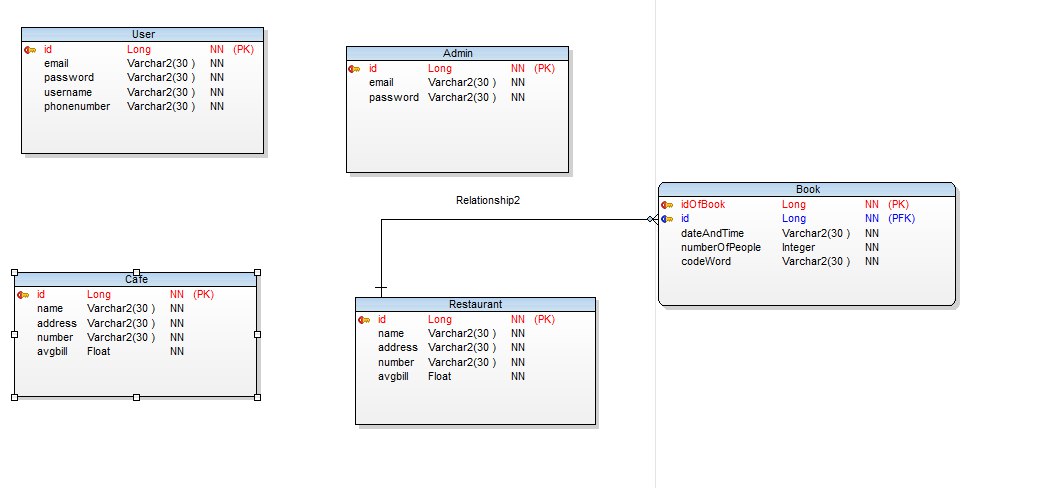


Таблица User содержит информацию об объектах сущности «пользователь». В таблице 1 приведены описания ее полей.

|  |  |  |
| --- | --- | --- |
| Таблица 1 – сущность «пользователь» | | |
| id | Long | Уникальный идентификатор объекта |
| email | Varchar2(30) | Почта пользователя |
| password | Varchar2(30) | Пароль пользователя |
| username | Varchar2(30) | Имя пользователя |
| phonenumber | Varchar2(30) | Мобильный телефон пользователя |

Таблица Admin содержит информацию об объектах сущности «администратор». В таблице 2 приведены описания ее полей.

|  |  |  |
| --- | --- | --- |
| Таблица 2 – сущность «администратор» | | |
| id | Long | Уникальный идентификатор объекта |
| email | Varchar2(30) | Почта администратора |
| password | Varchar2(30) | Пароль администратора |

Таблица Cafe содержит информацию об объектах сущности «кафе». В таблице 3 приведены описания ее полей.

|  |  |  |
| --- | --- | --- |
| Таблица 3 – сущность «кафе» | | |
| id | Long | Уникальный идентификатор объекта |
| name | Varchar2(30) | Название кафе |
| address | Varchar2(30) | Адрес кафе |
| number | Varchar2(30) | Номер телефона кафе |
| avgbill | Float | Средний чек в кафе |

Таблица Restaurant содержит информацию об объектах сущности «ресторан». В таблице 4 приведены описания ее полей.

|  |  |  |
| --- | --- | --- |
| Таблица 4 – сущность «ресторан» | | |
| id | Long | Уникальный идентификатор объекта |
| name | Varchar2(30) | Название ресторана |
| address | Varchar2(30) | Адрес ресторана |
| number | Varchar2(30) | Номер телефона ресторана |
| avgbill | Float | Средний чек в ресторане |

Таблица Book содержит информацию об объектах сущности «бронирование». В таблице 5 приведены описания ее полей.

|  |  |  |
| --- | --- | --- |
| Таблица 5 – сущность «бронирование» | | |
| idOfBook | Long | Уникальный идентификатор объекта |
| id | Long |  |
| dateAndTime | Varchar2(30) | Дата и время брони |
| numberOfPeople | Integer | Количество человек, на которых оформляется бронь |
| codeWord | Varchar2(30) | Кодовое слово для брони |

Начало формы

Конец формы

# 

# Глава 3. Организационная часть

Цель присутствия в интернете – предоставление возможности совершать бронирование онлайн, что упрощает жизнь клиентам и в то же время привлечет новых клиентов для заведений, расположенных у нас в каталоге. Так же у пользователей будем возможность увидеть перечень заведений, представленных в каталоге, а также можно воспользоваться поиском.

Основной целью приложения является снижение затрат на общение с клиентами. Именно это и является главной отличительной особенностью нашего продукта от продукта конкурента – Relax.by.

Выгоднее всего будет вкладывать деньги в развитие нашей разработки через Интернет. Все больше людей практически не выпускают из рук телефоны и не представляют свою жизнь без социальных сетей, поэтому такие платформы как Instagram, VKи т.п. будут идеальным местом, для размещения рекламы. Так же не секрет, что к мнению блогеров сейчас прислушиваются, так что идеальным вариантом было бы работать именно с популярным представителем в Интернет пространстве. В нашем случае отличным амбассадором стал бы фудблогер, который ведет аккаунт koko\_by. Как показывает практика, товары, которые просто мелькают у него в историях, потом недели 2 сложно найти на полках магазинов, а рестораны, о которых он публикует статьи, закрываются раньше положенного из-за того, что заканчиваются продукты, т.к. большой ажиотаж.

Так же в будущем можно будет реализовать рассылку на почту. Так пользователи будут получат информацию о новых заведениях, которые были добавлены в каталог, информацию о скидках, например, у нас несколько раз в году в Минске проводят gastrofest, в рамках которого заведения (участники данного вероприятия) предоставляют определенные сеты со скидкой. Мы могли бы принять некоторое участие в данном мероприятие, начать сотрудничать с организаторами gastrofest и, например, добавить раздел у себя на странице, посвященный ему. А также можно было бы высылать по e-mail рассылки информацию про данное мероприятие. Так мы засветимся на столь популярном ивенте, привлечем новых пользователей, т.е. это будет своего рода реклама для нас.

Таким образом, если грамотно провести рекламу, владельцы заведений будут сами платить нам за то, чтобы из заведения были у нас в каталоге. А также наша главная страница может стать своего рода платформой для рекламы, которая также будет приносить заработок. Но также не стоит забывать и о существование социальных сетей. Когда будут реализованы страницы в социальных сетях, посвященные нашему проекту, они станут приносить также дополнительный доход, т.к. заведения будут платить за то, чтобы на странице был опубликован пост или даже за обычную историю в Instagram.

# Заключение

В итоге в рамках данной курсовой работы были проделаны следующие действия:

* 1. Был спроектирован веб-интерфейс
* 2. Была создана база данных для хранения информации о заведениях (кафе и ресторанах), а также информация, введенная пользователем (бронирование и профиль)
* 3. Был реализован алгоритм поиска заведений при помощи языка Java и веб-файлов (.ftlh)
* 4. Была создана отдельная веб-страница профиля для более удобного использования веб-приложения пользователем (мониторинга бронирований)
* 5. Для каждого заведения реализована своя страница с полной информацией (описанием, номером телефона и т.д.) с возможностью подключения Google Maps
* 6. Реализована система авторизации и регистрации с подтверждением через почтовые сервисы (mail, gmail и т.д.)

Слабым местом данной системы является возможное занесение почтовый системы от Яндекс в спам из-за большого количества попыток регистрации, что сделает регистрацию на сайте невозможно, и придется перезапускать сервис. Помимо этого, на сайте не предусмотрена качественное разбиение бронирований на временные промежутки.

Основной составляющей системы является возможность просмотра каталога кафе и ресторанов и в последствие бронирование одного из них. Просмотр и отправка всей информации реализована при помощи .ftlh файлов. Существенная часть программа, т.е. обработка информации, реализована при помощи такого фреймворка как Spring. Для хранения информации использовались базы данных Oracle.

Самой сложной в плане реализации частью проекта является получение данных введенных пользователем или администратором с веб-страницы (обработка корректности введенных данных), а также обратное, т.е. отправка данных с приложения на веб-страницу. В случае запуска приложения на другом устройстве необходима корректировка некоторых полей (uploadpath), т.к. использовался абсолютный путь.

# Список использованных источников

* Уоллс К. Spring в действии – М.: ДМК Пресс, 2013. – 30 с.
* Блинов И.Н. Java Методы программирования: уч-мет. пособие / И.Н. Блинов, В.С. Романчик – М.: «Четыре четверти», 2013. – 14 с.
* Эккель Б. Философия Java – Спб.: Питер, 2009. – 39 с.
* https://javarush.ru/groups/posts/spring-framework-java-1
* <http://bourabai.ru/dbt/oracle.pdf>
* https://spring.io/guides