**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информатики и вычислительной техники

Кафедра ИиСП

**Отчёт по курсовой работе по дисциплине**

**«Разработка интернет приложений»**

Выполнил:

студент

группы ПС-41

Колчин И.А.

Проверил:

Габдулин Д. И.

г. Йошкар-Ола

2017

Оглавление

[Введение 3](#_Toc503702200)

[Задания лабораторных работ 4](#_Toc503702201)

[Структура программы 5](#_Toc503702202)

[Описание работы программы 6](#_Toc503702203)

[Ссылка на код лабораторных работ 7](#_Toc503702204)

[Использованные инструменты 7](#_Toc503702205)

# Введение

В качестве курсовой работы по предмету «Разработка интернет приложений» нужно было сделать интернет приложение работающие с базой данных. Курсовая работа была разделена на части в виде лабораторных работ. Каждая из лабораторных работ включает предыдущую, поэтому дальше будет описана структура последней лабораторной.

# Задания лабораторных работ

**1 лабораторная:**

1. Выбрать предметную область (книги, машины, одежда)
2. Определить атрибуты для сущности предметной области (цвет, размер, название и т.д.)
3. Создать форму добавляющую новую сущность
4. Вывести список внесенных сущностей

**2 лабораторная:**

Необходимо зафиксировать поведение разрабатываемого приложения с помощью 5 функциональных тестов.

Тесты должны проверять следующие ситуации:

* Введенные данные имеют различные типы - строковый, числовой и даты
* Полное заполнение формы валидными значениями
* Использование пустых значений и очень больших значений.
* Ввод невалидных данных

**3 лабораторная:**

* Организовать хранение сущностей в базе данных NuoDB
* Реализовать простейший шаблонизатор
* Вынести модель данных, бизнес логику и представление из Servlet’а
* Добавить логирование событий и ошибок

# Структура программы

В качестве предметной области были выбраны книги.

Программа состоит из следующих классов

* **MainServlet**

Обрабатывает запросы от пользователя, проверяет данные и генерирует страницы.

* **HtmlTemplater**

Шаблонизатор из 3 лабы, его задача генерировать страницу на основе шаблона и переданных переменных.

* **TemplateVariable**

Переменная для шаблонизатора, хранит имя и объект, который будет использоваться шаблонизатором.

* **BookDatabase**

Отвечает за работу с базой данных, позволяет добавлять, удалять данные.

* **Book**

В качестве предметной области были выбраны книги. Книга содержит в себе автора, название, количество страниц, дату издания.

При запуске приложения сначала открывается страница добавления книги. После загрузки страницы пользователь может добавить новую книгу или посмотреть список книг.

# Описание работы программы

Далее будет описано взаимодействие классов на примере действий пользователя и принцип работы шаблонизатора.

**Добавление книги:**

1. Отправка запроса с данными для новой книги классу **MainServlet**
2. Проверка данных
3. **MainServlet** формирует книгу(класс **Book**) и передаёт **BookDatabase** для добавления в базу
4. Если не все данные введены или введены не корректно то появится сообщение в верхней части окна. Для этого шаблонизатору(**HtmlTemplater**) будет дана команда сгенерировать страницу добавления книги с сообщением об ошибке. Перед генерацией шаблонизатору будет передано сообщение об ошибке, а затем дана команда для генерации страницы.
5. Если всё введено правильно то страница перезагрузится и вернётся в изначальное состояние. Шаблонизатор создаст новую страницу.

**Просмотр списка книг:**

1. Отправка запроса с данными для новой книги классу **MainServlet**
2. Сервлет запросит книги у **BookDatabase** и передаст шаблонизатору(**HtmlTemplater**).
3. Затем шаблонизатор сгенерирует страницу со списком

**Принцип работы шаблонизатора:**

1. Сначала передаются переменные, которые будут использоваться для генерации страницы с помощью команды **setAttribute(String name, String type, Object object),** где **name** – имя переменной, **type –**тип, **object**  - объект которые нужно передать.
2. Внутри шаблонизатора при выполнении **setAttribute** в список переменных вносится переменная в виде класса **TemplateVariable** куда передаются полученные аргументы.
3. Далее при генерации страницы название переменной используется для поиска в списке с **TemplateVariable**. Затем проверяется тип и производится конвертация содержимого переменной в строку для отображения.

# Ссылка на код лабораторных работ

https://github.com/7kia/PIA

# Использованные инструменты

1. Язык Java
2. База данных NuoDB
3. Сервер Tomcat