**Техническое задание на создание клиент-серверного веб приложения.**

**Интернет - кафе**. Пользователь зайдя на сайт, может выбрать сорта кофе, которые он хочет заказать, отметив соответствующие строки в списке возможных сортов. Список сортов храниться в базе данных. В списке на UI не показываются сорта с disabled='Y' (см. скрипт для создание БД). Для каждого сорта кофе пользователь может ввести количество чашек.

После того как пользователь определился с выбором, он нажимает кнопку "заказать" и вводит в появившейся странице свой адрес, на который необходимо доставить заказ и имя. После окончательного подтверждения, информация о заказе сохраняется в БД.

Правило подсчёта цены заказа:

1. каждая n чашка кофе одного сорта бесплатна.

Пример (n=5): 2 чашки чёрного кофе по 1 TGR и 4 чашки со сливками по 2 TGR в сумме стоят 10 TGR, но 6 чашек чёрного кофе по 1 TGR стоят 5 TGR, потому что 5 чашка бесплатная.

1. если заказ был сделан более чем на x TGR, то доставка бесплатна, в противном случае стоимость доставки = m TGR

Значения для n, x, m задаются в базе данных в таблице Configuration: id = 'n', id = 'x' и id = 'm' соответственно. Значения по умолчанию: n = 5, x = 10, m = 2

**Реализовано.** С использованием только: Spring MVC, JSP, JPA/Hibernate, jUnit, Spring IoC.

**Описание сборки проекта.** Проект реализован, как многомодульный maven проект. Модули:

* coffee-04-dao
* coffee-04-services
* coffee-04-web

До сборки проекта нужно создать базы данных – рабочую coffee и для тестирования – coffee\_test. Скрипты для создания баз данных лежат в каталоге [classPath]\script

Для доступа к базам данных в описании <**persistence-unit>** в

свойстах <**property name="javax.persistence.jdbc.user" value="root"**/> и <**property name="javax.persistence.jdbc.password" value="root"**/>

указать **login** и **password** (в настоящий момент указаны **root** и **root** соответственно).

Для рабочей базы coffee в файле [classPath]\coffee-04-dao\src\main\resources\META-INF\persistence.xml

Для тестовой базы coffee\_test в файлах [classPath]\coffee-04-dao\src\test\resources\META-INF\persistence.xml и [classPath]\coffee-04-services \src\test\resources\META-INF\persistence.xml

А так же в описании бина <**bean id="dataSource" class="org.apache.commons.dbcp2.BasicDataSource"**> в свойствах <**property name="username" value="root"**/> и <**property name="password" value="root"**/>

указать **login** и **password** (в настоящий момент указаны **root** и **root** соответственно).

Для рабочей базы coffee в файле [classPath]\coffee-04-dao\src\main\resources\beans-dao.xml

Для тестовой базы coffee\_test в файлах [classPath]\coffee-04-dao\src\test\resources\ testContext-dao.xml и [classPath]\coffee-04-services\src\test\resources\testContext- services.xml

Для сборки проекта перейти в папку [classPath], где содержится pom.xml проекта и запустить команду: mvn clean package –PtestCoverage

Результат работы плагина cobertura-maven-plugin, а именно отчет о степени покрытия приложения тестами, можно посмотреть в каждом модуле в файлах: [classPath]\[modulName]\target\site\cobertura\\*.html

При успешном прохождении тестов и сборки проекта будет сформирован war архив: [classPath]\target\coffee.war

При развертывании приложения на сервере Tomcat следует поместить файл coffee.war в директорию CATALINA\_HOME\webapps и при запуске Tomcat архив будет автоматически распакован и приложение станет доступно по контексту /coffee