ОГЛАВЛЕНИЕ

[1.1 Задание. Контрольная работа 1 1](#_Toc410914954)

[1.2 Задание. Контрольная работа 2. 1](#_Toc410914955)

[1.3 Задание. Курсовая работа 1. 2](#_Toc410914956)

[1.4 Список моделей программных систем. 2](#_Toc410914957)

## Задание. Контрольная работа 1

1. Выбрать модель программной системы из пункта 27.4.
2. На основе сущностей предметной области создать классы их описывающие.
3. Информацию о предметной области хранить в БД.
4. В качестве СУБД использовать MySQL.
5. Создать многомодульный проект Maven для выбранной модели программной системы.
6. Разбить проект на 3 модуля: **dao, services, web.**
7. Для авторизации и аутентификации использовать класс Filter.
8. Классы и методы должны иметь отражающую их функциональность названия и должны быть грамотно структурированы по пакетам.
9. Приложение должно соответствовать шаблону Model-View-Controller.
10. Для получения объектов dao и service использовать шаблон Singletone.
11. В web модуле для обработки запросов использовать шаблон Command.
12. Конфигурационную информацию хранить в properties-файле, например, такую как: параметры соединения с БД, граничные значения предметной области и сервера приложений (рекомендуется Tomcat)
13. При разработке использовать журналирование событий (Log4j).
14. Используя сервлеты и JSP, реализовать функциональности, предложенные в постановке конкретной задачи.
15. В JSP-страницах использовать возможности библиотеки JSTL.
16. Приложение должно поддерживать работу с кириллицей, в том числе и при хранении информации в БД.
17. При написании кода руководствоваться Java Code Convention.
18. 30% кода модулей dao и services покрыть unit-тестами.
19. 30% кода покрыть комментариями.

## Задание. Контрольная работа 2.

1. Использовать контрольную работу №1 изменить модуль dao. Переписать его с использованием фреймворка Hibernate.
2. Использовать отношения между сущностями 1-1, 1-n, n-n.
3. Ограничить количество элементов выборки с помощью педжинации.
4. Для запросов использовать Hibernate Query и Criteria.
5. Использовать механизм пользовательских транзакций.
6. Подключить кэш второго уровня Hibernate.
7. 60% кода модулей dao и services покрыть unit-тестами.

## Задание. Курсовая работа 1.

1. Использовать контрольную работу №2.
2. Для проекта произвести интеграцию Hibernate и Spring.
3. В качестве представления данных использовать шаблоны фрейморка Tiles.
4. Использовать модуль безопасности Spring.
5. Реализовать интернационализацию сообщений через Spring.

## Список моделей программных систем.

1. Система **Факультатив**. **Преподаватель** объявляет запись на **Курс**. **Студент** записывается на **Курс**, обучается и по окончании **Преподаватель** выставляет **Оценку**, которая сохраняется в **Архиве**. **Студентов**, **Преподавателей** и **Курсов** при обучении может быть несколько.
2. Система **Платежи**. **Клиент** имеет **Счет** в банке и **Кредитную** **Карту (КК)**. **Клиент** может оплатить **Заказ**, сделать платеж на другой **Счет**, заблокировать **КК** и аннулировать **Счет**. **Администратор** может заблокировать **КК** за превышение кредита.
3. Система **Вступительные экзамены**. **Абитуриент** регистрируется на **Факультет**, сдает **Экзамены**. **Преподаватель** выставляет **Оценку**. Система подсчитывает средний балл и определяет **Абитуриентов**, зачисленных в учебное заведение.
4. Система **Библиотека**. **Читатель** оформляет **Заказ** на **Книгу**. Система осуществляет поиск в **Каталоге**. **Библиотекарь** выдает **Читателю** **Книгу** на абонемент или в читальный зал. При невозвращении **Книги** **Читателем** он может быть занесен **Администратором** в “черный список”.
5. Система **Телефонная станция**. **Абонент** оплачивает **Счет** за разговоры и **Услуги**, может попросить **Администратора** сменить номер и отказаться от услуг. **Администратор** изменяет номер, **Услуги** и временно отключает **Абонента** за неуплату.
6. Система **Автобаза**. **Диспетчер** распределяет заявки на **Рейсы** между **Водителями** и назначает для этого **Автомобиль**. **Водитель** может сделать заявку на ремонт. **Диспетчер** может отстранить **Водителя** от работы. **Водитель** делает отметку о выполнении **Рейса** и состоянии **Автомобиля**.
7. Система **Интернет-магазин**. **Администратор** добавляет информацию о **Товаре**. **Клиент** делает и оплачивает **Заказ** на **Товары**. **Администратор** регистрирует **Продажу** и может занести неплательщиков в “черный список”.
8. Система **Городской транспорт**. На **Маршрут** назначаются **Автобус**, **Троллейбус** или **Трамвай**. Транспортные средства должны двигаться с определенным для каждого **Маршрута** интервалом. При поломке на **Маршрут** должен выходить резервный транспорт или увеличиваться интервал движения.
9. Система **Аэрофлот**. **Администратор** формирует летную **Бригаду** (пилоты, штурман, радист, стюардессы) на **Рейс**. Каждый **Рейс** выполняется **Самолетом** с определенной вместимостью и дальностью полета. **Рейс** может быть отменен из-за погодных условий в **Аэропорту** отлета или назначения. **Аэропорт** назначения может быть изменен в полете из-за технических неисправностей, о которых сообщил командир.
10. Система **Периодические издания**. **Читатель** может сделать **Заявку**, предварительно выбрав периодические **Издания** из списка. Система подсчитывает сумму для оплаты. **Читатель** оплачивает **заявку**. **Администратор** добавляет **Заявку** в «**чёрный** **список**», если **Клиент** не оплачивает её в определённый срок.
11. Система **Заказ гостиницы**. **Клиент** оставляет **Заявку** на **Номер**. Указав количество мест в номере, класса апартаментов и времени пребывания. **Администратор** рассматривает **Заявку**, подтверждает или отклоняет. Результат просматривает **Клиент**. В случае подтверждения **Заявки** **Клиент** оплачивает услуги.
12. Система **Прокат** **автомобилей**. **Клиент** выбирает **Автомобиль** из списка доступных. Заполняет форму **Заказа**, указывая паспортные данные, срок аренды. **Администратор** может отклонить **Заявку**, указав причины отказа. При подтверждении **Заявки**, **Клиент** оплачивает **Заказ**. Система выписывает сумму. В случае повреждения **Автомобиля** **Клиентом**, **Администратор** вносит соответствующие пометки.
13. Система **Новостной портал**. **Пользователь** может читать **Новости**. Авторизированный **Пользователь** может создавать свои **Новости** и редактировать их, а так же оставлять **Комментарии** к другим новостям**.** **Модератор** может удалять и редактировать **Новости** и **Комментарии**.