

Лабораторная работа №4 (0)

Построить веб-систему, поддерживающую заданную функциональность:

- На основе сущностей предметной области создать классы их описывающие.
 - Классы и методы должны иметь отражающую их функциональность названия и должны быть грамотно структурированы по пакетам.
 - Оформление кода должно соответствовать Java Code Convention.
 - Информацию о предметной области хранить в БД, для доступа использовать API JDBC с использованием пула соединений, стандартного или разработанного самостоятельно. В качестве СУБД рекомендуется MySQL или Oracle.
 - Приложение должно поддерживать работу с кириллицей (быть многоязычной), в том числе и при хранении информации в БД.
 - Архитектура приложения должна соответствовать шаблону Model-View-Controller.
 - При реализации алгоритмов бизнес-логики использовать шаблоны GoF: Factory Method, Command, Builder, Strategy, State, Observer etc.
 - Используя сервлеты и JSP, реализовать функциональности, предложенные в постановке конкретной задачи.
 - В страницах JSP применять библиотеку JSTL и разработать собственные теги.
 - При разработке бизнес логики использовать сессии и фильтры.
 - Выполнить журналирование событий, то есть информацию о возникающих исключениях и событиях в системе обрабатывать с помощью Log4j.
 - Код должен содержать комментарии.
1. Система **Факультатив**. Существует перечень **Курсов**, за каждым из которых закреплен один **Преподаватель**. **Студент** записывается на один или несколько **Курсов**. По окончании обучения **Преподаватель** выставляет **Студенту** и добавляет отзыв.
 2. Система **Платежи**. **Клиент** имеет одну или несколько **Кредитных Карт**, каждая из которых соответствует некоторому **Счету** в системе платежей. **Клиент** может при помощи **Счета** сделать **Платеж**, заблокировать **Счет** и пополнить **Счет**. **Администратор** снимает блокировку.
 3. Система **Больница**. **Врач** определяет диагноз, делает назначение **Пациенту** (процедуры, лекарства, операции). Назначение может выполнить **Медсестра**

(процедуры, лекарства) или **Врач** (любое назначение). **Пациент** может быть выписан из **Больницы**, при этом фиксируется окончательный диагноз.

4. Система **Турагентство**. **Заказчик** выбирает и оплачивает **Тур** (отдых, экскурсия, шоппинг). **Турагент** определяет тур как «горящий», размеры скидок постоянным клиентам.
5. Система **Телефонная станция**. **Администратор** осуществляет подключение **Абонентов**. **Абонент** может выбрать одну или несколько из предоставляемых **Услуг**. **Абонент** оплачивает **Счет** за разговоры и **Услуги**. **Администратор** может просмотреть список неоплаченных **Счетов** и заблокировать **Абонента**.
6. Система **Автобаза**. **Диспетчер** распределяет **Заявки** на **Рейсы** между **Водителями**, за каждым из которых закреплен свой **Автомобиль**. На **Рейс** может быть назначен **Автомобиль**, находящийся в исправном состоянии и характеристики которого соответствуют **Заявке**. **Водитель** делает отметку о выполнении **Рейса** и состоянии **Автомобиля**.
7. Система **Интернет-магазин**. **Администратор** осуществляет ведение каталога **Товаров**. **Клиент** делает и оплачивает **Заказ** на **Товары**. **Администратор** может занести неплательщиков в «черный список».
8. Система **Авиакомпания**. **Авиакомпания** имеет список рейсов. **Диспетчер** формирует летную **Бригаду** (пилоты, штурман, радист, стюардессы) на **Рейс**. **Администратор** управляет списком рейсов.
9. Система **Заказ гостиницы**. **Клиент** заполняет **Заявку**, указывая количество мест в номере, класс апартаментов и время пребывания. **Администратор** просматривает поступившую **Заявку**, выделяет наиболее подходящий из доступных **Номеров**, после чего система выставляет **Счет Клиенту**.
10. Система **Жилищно-коммунальные услуги**. **Квартиросъемщик** отправляет **Заявку**, в которой указывает род работ, масштаб, и желаемое время выполнения. **Диспетчер** формирует соответствующую **Бригаду** и регистрирует её в **Плане работ**.
11. Система **Ресторан**. **Клиент** осуществляет заказ из **Меню**. **Администратор** подтверждает **Заказ** и отправляет его на кухню для исполнения. **Администратор** выставляет **Счет**. **Клиент** производит его оплату.
12. Система **Кофе-машина**. **Пользователь** обладает **Счетом**. **Кофе-машина** содержит набор **Напитков**, с заданным числом порций и дополнительных **Ингредиентов**. **Пользователь** может купить один или несколько **Напитков**. **Администратор Кофе-машины** осуществляет ее наполнение.
13. Система **Парк**. **Владелец** парка дает указания **Леснику** о высадке (лечении, художественной обработке, уничтожении) **Растений**. **Лесник** отчитывается о выполнении. **Владелец** просматривает результаты и подтверждает исполнение.
14. Система **Команда разработчиков**. **Заказчик** представляет **Техническое Задание (ТЗ)**, в котором перечислен перечень **Работ** с указанием

квалификации и количества требуемых специалистов. **Менеджер** рассматривает **ТЗ** и оформляет **Проект**, назначая на него незанятых **Разработчиков** требуемой квалификации, после чего рассчитывается стоимость **Проекта** и **Заказчику** выставляется **Счет**. **Разработчик** имеет возможность отметить количество часов, затраченных на работу над проектом.

15. Система **Городской транспорт**. На **Маршрут** назначаются **Автобус**, **Троллейбус** или **Трамвай**. Транспортные средства должны двигаться с определенным для каждого **Маршрута** интервалом или расписанием.