Построить веб-систему, поддерживающую заданную функциональность:

1. На основе сущностей предметной области создать классы, их описывающие;
2. Классы и методы должны иметь названия отражающие их функциональность, и быть грамотно структурированы по пакетам;
3. Оформление кода должно соответствовать Java Code Convention;
4. Информацию о предметной области хранить в БД, для доступа использовать API JDBC с использованием пула соединений, стандартного или разработанного самостоятельно. В качестве СУБД использовать MySQL;
5. Приложение должно поддерживать работу с кириллицей, в том числе и при хранении информации в БД;
6. Архитектура приложения должна соответствовать шаблону Model-View-Controller;
7. При реализации алгоритмов бизнес-логики использовать шаблоны GoF и GRASP;
8. Используя сервлеты и JSP, реализовать функции, предложенные в постановке конкретной задачи;
9. В страницах JSP применять библиотеку JSTL;
10. При разработке бизнес-логики использовать сессии и фильтры;
11. Код должен содержать комментарии.

Задание 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.