В самом конце есть условия семестровой работы, а задания с самого начала дока (1 по 12) я хочу вставить в эту семестровую работу. Идея сайта в следующем: я деаю сайт в котором ты регестрируешься и попадаешь а свой профиль. В профиле указываются твои данные и список видеоуроков(мною отснятых) по музыкальной грамотности. Под каждым видео хочу сделать возможность оставить комментарий(дабы по заданию семестровки нужно 2 сущности: у меня это юзер и комментарий).Прикреплю также вам весь проект что уже написал, он работает лишь на треть и то криво. потом задания с 13-15 это 2 семестровая работа(ее я почти доделал), а последующие где рисовалка (16-18) я сделал. меня так же беспокоят все дополнительные задания.

Домашки

1. Сверстать HTML страницу с несколькими абзацами, заголовками и подзаголовками (как в статьях/книгах). Сделать список ссылок, указывающих на соот.разделы документа.
2. Создать графический макет своего сайта

    2.1. Установить сервер приложений (Tomcat, GlassFish, jBoss или другой на выбор). Через Maven запустить и создать проект

* создать свой архетип на основе готового проекта, использовать его в другим проекте.

1. Запустить сервлет на котором будет форма регистраций (уделить особое внимание правильности типов полей!) ‒ email, пароль, страна (выбор), пол (выбор). Реализовать запись этих данных в какой-нибудь файл в человекочитеамом формате (csv, xml и проч).

* Проверять на корректный пароль и email

1. Добавить проверку регулярками и сделать правильную архитектуру регистрации (модульность, SOLID) и реализовать защиту от повторной отправки формы (исп.перенаправление)
2. Привести в соответствие шаблону проектирования MVС. Добавить использование JSP (с JSTL). Можно использовать другие шаблонизаторы (Freemaker, JTwig и др по согласованию с АА).
3. Добавить аутентификацию-авторизацию реализованную через сессии. В проекте должна быть страница (сервлет), доступ к которой есть только у авторизованных пользователей.
4. Прикрутить к сайту показ погоды своего города и количество прослушиваний у любимого исполнителя(API).  Установить СУБД.
5. Подключить БД, создать вторую сущность. Реализовать CRUD для одной из сущностей.
6. Проверить форму регистрации в браузере (JS), и не отправлять в случае неверного формата данных - подчеркнуть/выделить красным цветом там где ошибка, подписать, что заполнено не так. Все поля формы должны проверяться соответствующим им способом (проверка длины, использование регулярных выражений для проверки общего вида строки, проверка на заполнение поля и проч.).
7. Переделать проверку регистрации через JQuery. Реализовать регистрацию на  AJAX. Задание зачтётся и без второй части, но будут получены неполные баллы.
8. Реализовать REST-сервис с CRUD для любой сущности. Можно реализовать не на основе своего семестрового проекта.
9. +Написать универсальную ORM с CRUD-операциями. Использовать рефлексию. Система должна работать с любым классом-сущностью, у которого есть правильно именованные методы установки и получения (setVar, getVar), поле int id, все поля имеют числовые, строковые или Calendar-типы. ORM может быть заточена под конкретную СУБД, но работать должна обязательно через JDBC.
10. Сделать нормальный запрос к сайту КФУ с использованием Socket. Реализовать клиент и сервер на основе сокетов с элементарным общением (Например, “привет-привет”).
11. Создать многопоточный сервер с одной общей переменной между разными клиентами (соединениями). Доступ к переменной должен быть потокобезопасный.
12. Сделать многопоточную игру на сервере. Должен быть многопоточный сервер работающий с несколькими клиентами, каждый клиент обслуживается в отдельном потоке. Общий протокол общения клиент и сервер. Логика самого сервера(игры) - общие переменные, какой клиент должен отвечать по очереди. Хороший пример: игра в города, но её делать нельзя.
13. Окна, погода и проч.
14. Рисовалка:

* Изменение цвета кисти
* Изменение размера кисти
* Заливка цветом
* \*Заливка фона изображения(растянуть)
* \*Рисовать непрерывно и плавно

     18. Upgrade рисовалки

* Зеркальное отображение
* Поворот
* \*Анимация картинки, что бы крутилась вокруг одного из своих углов

  19. Реализовать метод Симпсона поиска приближённого значения определённого интеграла.

Дополнительные задания

1. Программа считывает из файла имена муз.исполнителей, перечисленных через запятую. Далее она делает запрос на Last.fm через его API и генерирует HTML-страницу со списком информации по всем исполнителям с отображением имени, числа прослушиваний и связанным с исполнителем изображением. Информация должна подаваться в читаемом виде с должным оформлением.
   1. Аналогично получить информацию об исполнителях и записать её в БД в соот.таблицу.

2. RFC 6455 перевести всю первую часть к среде <https://tools.ietf.org/html/rfc6455#section-1.1>

3.   Нарисовать на элементе окна SWING пятиконечную звезду по заданным размерам пользователя (расстояние от центра до кончиков).

4.   Написать универсальную ORM, с методами select, insert, которая получает название класса/таблицы и предоставляет возможность получить все записи таблицы, возвратить коллекцию экземпляров соот. класса сущности и вставить данные одного из экземпляров этой сущности в таблицу. Написать сопутствующие соответствующие исключения.

5. Поиск минимума функции на заданном интервале.

Контрольные работы

1. HTML, CSS
2. Swing+HTTP+REST

Семестровые работы

Сайт (сервлеты)

При проверке будет оцениваться соответствие всем нижеперечисленным требованиям, реализация правильной архитектуры, умение применять используемые технологии, удобство пользования системой, защищённость системы.

Требования:

1. Сайт, написанный на сервлетах без использования Java Web фреймворков.
2. У сайта должна быть идея, обладающая новизной и актуальностью.
3. Сайт должен иметь удобный интерфейс.
4. Необходимо использовать HTML5, CSS3.
5. Нужно использовать шаблонизатор: JSP (с JSTL), Velocity, JTWIG или другой. В коде видов нельзя напрямую вызывать Java-код.
6. Сайт должен иметь аутентификацию, авторизацию, регистрацию.
7. Сайт должен хранить данные в СУБД: MySQL, MSSQL, PostgreSQL. Можно ли использовать другие СУБД нужно уточнить у преподавателя.
8. Для работы с СУБД необходимо использовать JDBC (с PrepareStatement!).
9. Система должна работать, как минимум с двумя сущностями, которые связаны. Например, Пользователь и Сообщение на форуме.
10. Как минимум, для одной из сущностей должен быть реализован CRUD.
11. Система должна быть построена по модели MVC. Необходимо придерживаться принципов SOLID. Весь код должен быть корректно поделён на пакеты.
12. Обдуманное использование JavaScript, AJAX не обязательно, но положительно повлияет на оценку.
13. Проект должен быть основан на проекте Maven.

*После 9 ноября максиму за семестровую работу: 4 балла.*

Сетевая игра (дополнительно)

1. Программа должна иметь серверную и клиентскую части. Клиенты взаимодействуют через сервер, не напрямую.
2. Для реализации взаимодействия необходимо хорошо продумать протокол.
3. Программа должна быть игрой, в которую играет 2 или более человек. При этом сервер должен контролировать соблюдение правил, отсчитывать время хода, считать очки или совершать другие игровые действие, не позволяющие играть нечестно в контексте правил игры.
4. Игра не должна быть примитивной: это не может быть игра города или простые крестики нолики. Если есть сомнения, лучше уточните заранее, можно ли реализовывать ту или иную игру. Если игра будет слишком простой, возможно, её и можно будет сдать, но с дополнением системы чатом.
5. Серверная часть должна быть многопоточной: обрабатывать одновременно запросы нескольких клиентов. При этом все процессы должны быть потокобезопасны.
6. Клиент должен иметь оконный интерфейс, написанный на SWING.
7. У клиента хотя бы один элемент должен иметь свою логику отрисовки (см. метод paintComponent). Это может быть карта, это могут быть часы или другая анимация, связанная с самой игрой.
8. (Дополнительно) Реализовать комнаты/лобби - возможность клиентам играть друг с другом, не мешая другим, организуя сессии игр.

*Семестровая работа не является обязательной. Для того, чтобы получить соот.срокам баллы, нужно к этим срокам сдать все доп.штрафные задания.*

*Баллы: За неё можно получить 7 дополнительных к семестровым баллов, если сдать все пункты 1-7; и ещё 3 балла за пункт 8.*

*Сроки:*

1. *До 14.12.16 включительно -  можно получить ещё +2 балла.*
2. *До 16.12.16 включительно - можно получить 7+3 балла*
3. *Позже - 4 + 3 балла.*

*Таким образом, можно получить до 12 баллов, есть сдать семестровую работу до 14.12.16, выполнив 8 пунктов,*