Модуль EJB

* Пакет entities (содержит все сущности ):
* Client

Содержит методы get и set для своих полей

* Goal

Содержит методы get и set для своих полей

* GoalUser

Содержит методы get и set для своих полей

* Level

Содержит методы get и set для своих полей

* Message

Содержит методы get и set для своих полей

* Personage

Содержит методы get и set для своих полей

* Topic

Содержит методы get и set для своих полей

* User

Содержит методы get и set для своих полей

* UserRole

Содержит методы get и set для своих полей

* UserRolePK

Содержит методы get и set для своих полей

* Пакет model
* AbstractFacade – абстрактный класс
* public void create(T entity) – на вход приходит объект сущности, создается новая запись в базе данных
* public void edit(T entity) – на вход приходит объект сущности, обновляется запись в базе данных
* public void remove(T entity) – на вход приходит объект сущности, удаляется запись из базы данных
* public T find(Object id) – на вход приходит id, возвращается объект с соответствующим id
* public List<T> findAll() – вывод всех записей из таблицы БД
* ClientFacade – класс, соответствующий сущности client из базы данных, наследует AbstractFacade, реализует ClientFacadeLocal
* ClientFacadeLocal – локальный интерфейс, содержит методы:
* public void create(Client client) – на вход приходит объект сущности, создается новая запись в базе данных
* public void edit(Client client) – на вход приходит объект сущности, обновляется запись в базе данных
* public void remove(Client client) – на вход приходит объект сущности, удаляется запись из базы данных
* public T find(Object id) – на вход приходит id, возвращается объект с соответствующим id
* public List<Client> findAll() – вывод всех записей из таблицы БД
* GoalFacade - класс, соответствующий сущности goal из базы данных, наследует AbstractFacade, расширяет GoalFacadeLocal
* GoalFacadeLocal – локальный интерфейс, содержит методы:
* public void create(Goal goal) – на вход приходит объект сущности, создается новая запись в базе данных
* public void edit(Goal goal) – на вход приходит объект сущности, обновляется запись в базе данных
* public void remove(Goal goal) – на вход приходит объект сущности, удаляется запись из базы данных
* public T find(Object id) – на вход приходит id, возвращается объект с соответствующим id
* public List<Goal> findAll() – вывод всех записей из таблицы БД
* GoalUserFacade - класс, соответствующий сущности goal\_user из базы данных, наследует AbstractFacade, расширяет GoalUserFacadeLocal
* GoalUserFacadeLocal - локальный интерфейс, содержит методы:
* public void create(GoalUser goaluser) – на вход приходит объект сущности, создается новая запись в базе данных
* public void edit(GoalUser goaluser) – на вход приходит объект сущности, обновляется запись в базе данных
* public void remove(GoalUser goaluser) – на вход приходит объект сущности, удаляется запись из базы данных
* public T find(Object id) – на вход приходит id, возвращается объект с соответствующим id
* public List<GoalUser> findAll() – вывод всех записей из таблицы БД
* LevelFacade - класс, соответствующий сущности level из базы данных, наследует AbstractFacade, реализует LevelFacadeLocal
* LevelFacadeLocal - локальный интерфейс, содержит методы:
* public void create(Level level) – на вход приходит объект сущности, создается новая запись в базе данных
* public void edit(Level level) – на вход приходит объект сущности, обновляется запись в базе данных
* public void remove(Leevl level) – на вход приходит объект сущности, удаляется запись из базы данных
* public T find(Object id) – на вход приходит id, возвращается объект с соответствующим id
* public List<Level> findAll() – вывод всех записей из таблицы БД
* MessageFacade - класс, соответствующий сущности message из базы данных, наследует AbstractFacade, реализует MessageFacadeLocal
* MessageFacadeLocal - локальный интерфейс, содержит методы:
* public void create(Message message) – на вход приходит объект сущности, создается новая запись в базе данных
* public void edit(Message message )– на вход приходит объект сущности, обновляется запись в базе данных
* public void remove(Message message) – на вход приходит объект сущности, удаляется запись из базы данных
* public T find(Object id) – на вход приходит id, возвращается объект с соответствующим id
* public List<Message> findAll() – вывод всех записей из таблицы БД
* PersonageFacade - класс, соответствующий сущности personage из базы данных, наследует AbstractFacade, реализует PersonageFacadeLocal
* PersonageFacadeLocal - локальный интерфейс, содержит методы:
* public void create(Personage personage) – на вход приходит объект сущности, создается новая запись в базе данных
* public void edit(Personage personage)– на вход приходит объект сущности, обновляется запись в базе данных
* public void remove(Personage personage) – на вход приходит объект сущности, удаляется запись из базы данных
* public T find(Object id) – на вход приходит id, возвращается объект с соответствующим id
* public List< Personage> findAll() – вывод всех записей из таблицы БД
* TopicFacade - класс, соответствующий сущности topic из базы данных, наследует AbstractFacade, реализует TopicFacadeLocal
* TopicFacadeLocal - локальный интерфейс, содержит методы:
* public void create(Topic topic) – на вход приходит объект сущности, создается новая запись в базе данных
* public void edit(Topic topic)– на вход приходит объект сущности, обновляется запись в базе данных
* public void remove(Topic topic) – на вход приходит объект сущности, удаляется запись из базы данных
* public T find(Object id) – на вход приходит id, возвращается объект с соответствующим id
* public List<Topic> findAll() – вывод всех записей из таблицы БД
* UserFacade - класс, соответствующий сущности topic из базы данных, наследует AbstractFacade, реализует UserFacadeLocal
* UserFacadeLocal - локальный интерфейс, содержит методы:
* public void create(User user) – на вход приходит объект сущности, создается новая запись в базе данных
* public void edit(User user)– на вход приходит объект сущности, обновляется запись в базе данных
* public void remove(User user) – на вход приходит объект сущности, удаляется запись из базы данных
* public T find(Object id) – на вход приходит id, возвращается объект с соответствующим id
* public List<User> findAll() – вывод всех записей из таблицы БД
* UserRoleFacade - класс, соответствующий сущности topic из базы данных, наследует AbstractFacade, реализует UserRoleFacadeLocal
* UserRoleFacadeLocal - локальный интерфейс, содержит методы:
* public void create(UserRole userrole) – на вход приходит объект сущности, создается новая запись в базе данных
* public void edit(UserRole userrole)– на вход приходит объект сущности, обновляется запись в базе данных
* public void remove(UserRole userrole) – на вход приходит объект сущности, удаляется запись из базы данных
* public T find(Object id) – на вход приходит id, возвращается объект с соответствующим id
* public List<UserRole> findAll() – вывод всех записей из таблицы БД

Модуль WAR

* ForumBean – бин форума
* public void SendMessage() – метод отправки сообщений
* public String messageIdTopic()– получение списка сообщений по теме

обрабатываемое исключение: EJBException

* public List<Topic> getAllTopics() - получение списка тем

обрабатываемое исключение: EJBException

* public List<Message> getAllMessage() – получение списка сообщений

обрабатываемое исключение: EJBException

* public String deleteMessage(Message message) – удаление сообщения

обрабатываемое исключение: EJBException

* public String createTopic() – создание темы

обрабатываемое исключение: EJBException

* public String createMessage() – создание сообщения

обрабатываемое исключение: EJBException

* public String ban() – назначить или снять бан на форуме

обрабатываемое исключение: EJBException

* set и get методы для полей
* AdministratorBean:
* public List<User> getAllUser() – вывод списка пользователей
* public StreamedContent getImageCreate() – вывод только что добавленной картинки

обрабатываемое исключение: IOException

* public byte[] InputStreamToArryByte(InputStream in) – вывод картинки в байтовый массив

обрабатываемое исключение: IOException

* public void uploadImagePersonage() – добавление картинки персонажу на уровень

обрабатываемое исключение: EJBException

* public List<String> levelFild() – генерация списка уровней для добавления
* public String ban() – назначить или снять бан пользователя

обрабатываемое исключение: EJBException

* public String addAdmin() – назначение прав администратора

обрабатываемое исключение: EJBException

* public List<Goal> getAllGoal() – вывод общего списка целей
* public String directory() – добавление цели в список дефолтных

обрабатываемое исключение: EJBException, IOException

* public void uploadImageGoalDef() – добавление картинки дефолтной цели

обрабатываемое исключение: EJBException

* public String deleteGoal() – удаление цели

обрабатываемые исключения: EJBException

* set и get методы для полей
* ClientBean:
* public String editClient(Client c) – просмотр личных данных

обрабатываемые исключения: EJBException

* public List<GoalUser> findAllGoalCurrentClient() – вывод целей клиента

обрабатываемые исключения: EJBException

* public String createGoalDefoltUser() – добавление дефолтной цели

обрабатываемые исключения: EJBException, NumberFormatException

* public String editGoalUser() – метод перехода на конкретную цель

обрабатываемые исключения: EJBException, NumberFormatException

* public String deleteGoalUser() – удаление цели пользователя

обрабатываемые исключения: EJBException, NumberFormatException

* public String createGoalUser() – добавление новой цели пользователя

обрабатываемое исключение: EJBException, IOException

* public void findAllGoalDefolt() – поиск по дефолтным целям

обрабатываемое исключение: EJBException

* public List<Goal> AllGoalDefolt() – вывод дефолтных целей

обрабатываемое исключение: EJBException

* public String addLevel() – выполнение цели на текущий день

обрабатываемое исключение: EJBException

* public int percentСomplianceGoalUser() – просчитывается выхоленность цели

обрабатываемое исключение: EJBException

* public ScheduleModel getEventModel() – модель для календаря
* private void createAreaModel() – метод для прорисовки графика

обрабатываемое исключение: EJBException, EnumConstantNotPresentException

* public boolean checkDate() – метод скрытия кнопки

обрабатываемое исключение: EJBException

* public boolean isDateEqual(Date dateLevel) – метод выполняющий сравнение дат
* public entity.Client calculateClientReiting(entity.Client client) – рейтинг одного пользователя
* public List<entity.Client> calculateClientsRating() – метод который считает рейтинг для всех пользователей

обрабатываемое исключение: EJBException

* private int compaeInts(int a, int b) – компоратор для сортировки по рейтингу
* public boolean payment(String holder, String codeCard, String codeSecurity, String expirationDate, String purchaseValue) throws IOException – оплата персонажа
* public StreamedContent getStreamedImageById() – вывод картинки персонажу

обрабатываемое исключение: EJBException, NumberFormatException

* public StreamedContent getStreamedImageByIdGoalDefolt() – вывод картинок дефолтным целям

обрабатываемое исключение: EJBException, NumberFormatException

* public StreamedContent getStreamedImageByIdAndLevel() – вывод картинок по уровню

обрабатываемое исключение: EJBException, NumberFormatException

* public List<Personage> AllPersonages() – получение списка персонажей

обрабатываемое исключение: EJBException

* public byte[] InputStreamToArryByte(InputStream in) throws IOException – чтение картинки из потока в байтовый массив
* public void uploadProfile() – добавление фотографии на профиль пользователя

обрабатываемое исключение: EJBException, IOException

* public StreamedContent getAvatar() – вывод фотографии пользователя

обрабатываемое исключение: EJBException

* public List<Personage> personagesFild() – получения списка персонажей
* set и get методы для полей
* UserBean:
* public String createUser() – создание нового пользователя

обрабатываемое исключение: EJBException

* public User getCurrentUserObject() – вернуть текущего пользователя

обрабатываемое исключение: EJBException

* public String getCurrentUser() – получение логина текущего пользователя

обрабатываемое исключение: EJBException, NumberFormatException

* public String logout() – выйти из системы
* public boolean isUserLoggedIn() – проверка аутентифицирован ли пользователь