**Цель работы:** Написать простейшее веб-приложение с использованием технологии JSF, JPA.

**Ход Работы:**

1. Создал базу bankdb на сервере MySQL и подключение к ней.
2. Создал проект web-приложения с использованием JSF 2.2 (facelets).
3. Создал entity Account и persistence.xml.
4. В качестве источника данных использовал созданное подключение проименованное JNDI bank.
5. Дореализовал Account. Добавил поля данных ownername и balance, доопределил для них гетеры и сетеры, создал методы работы с аккаунтом.
6. Создал фасад для работы с сущностью Account.
7. Создал управляющий бин для моих jsf – страниц.
8. Реализовал форму на index.xhtml для ввода данных об аккаунте.
9. Создал страницу response.xhtml.
10. Подключил в мой бин фасад для аккаунта и функцию actionMethod.
11. При попытке запуска высветилась ошибка: bank\_\_pm ресурс не определен.
12. Разгадка ошибки крылась в glassfish-e. Необходимо было в консоли администратора создать пул соединений с MySQL и определить JDBC ресурс под JNDI именем bank, связанный с этим пулом, что я и сделал. После данных манипуляций приложение запустилось.

**Скриншоты:**

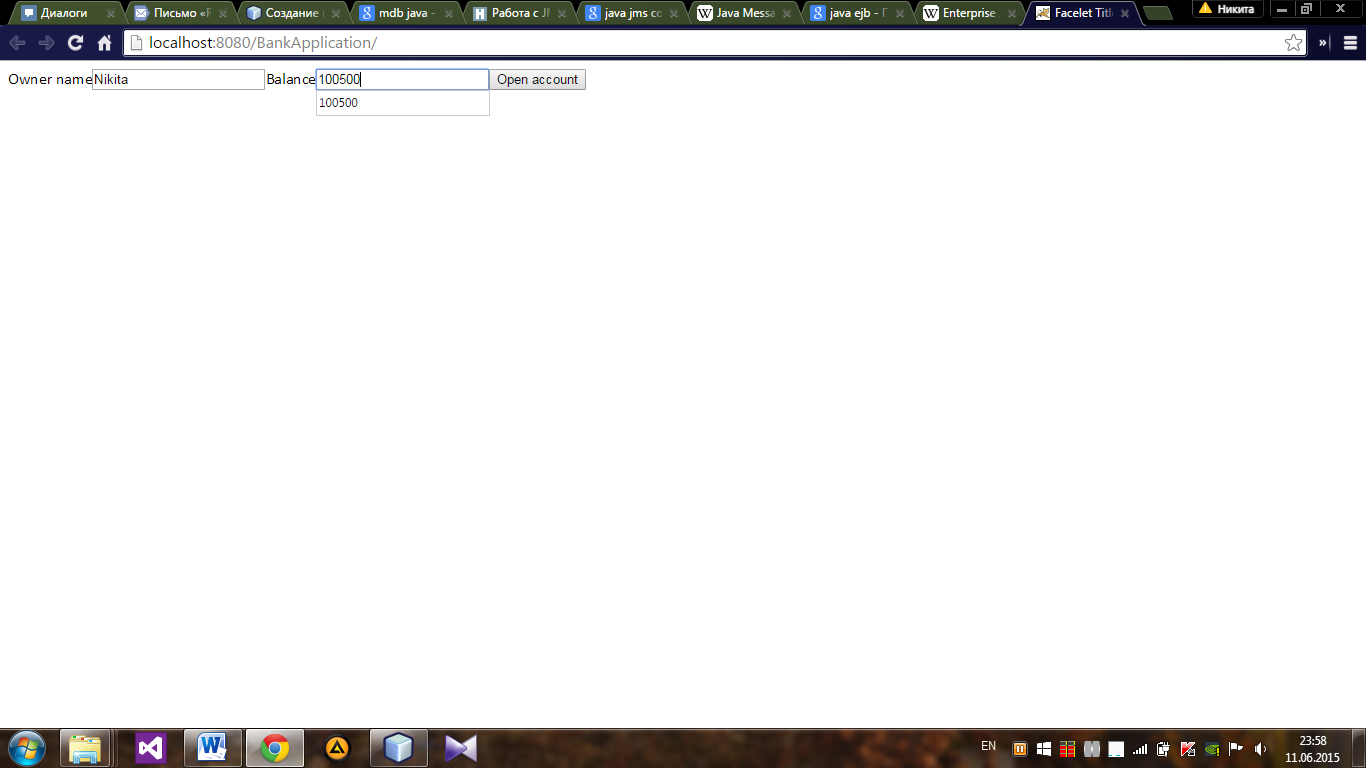


Рисунок 1 – Начальная страница приложения с заполненной формой

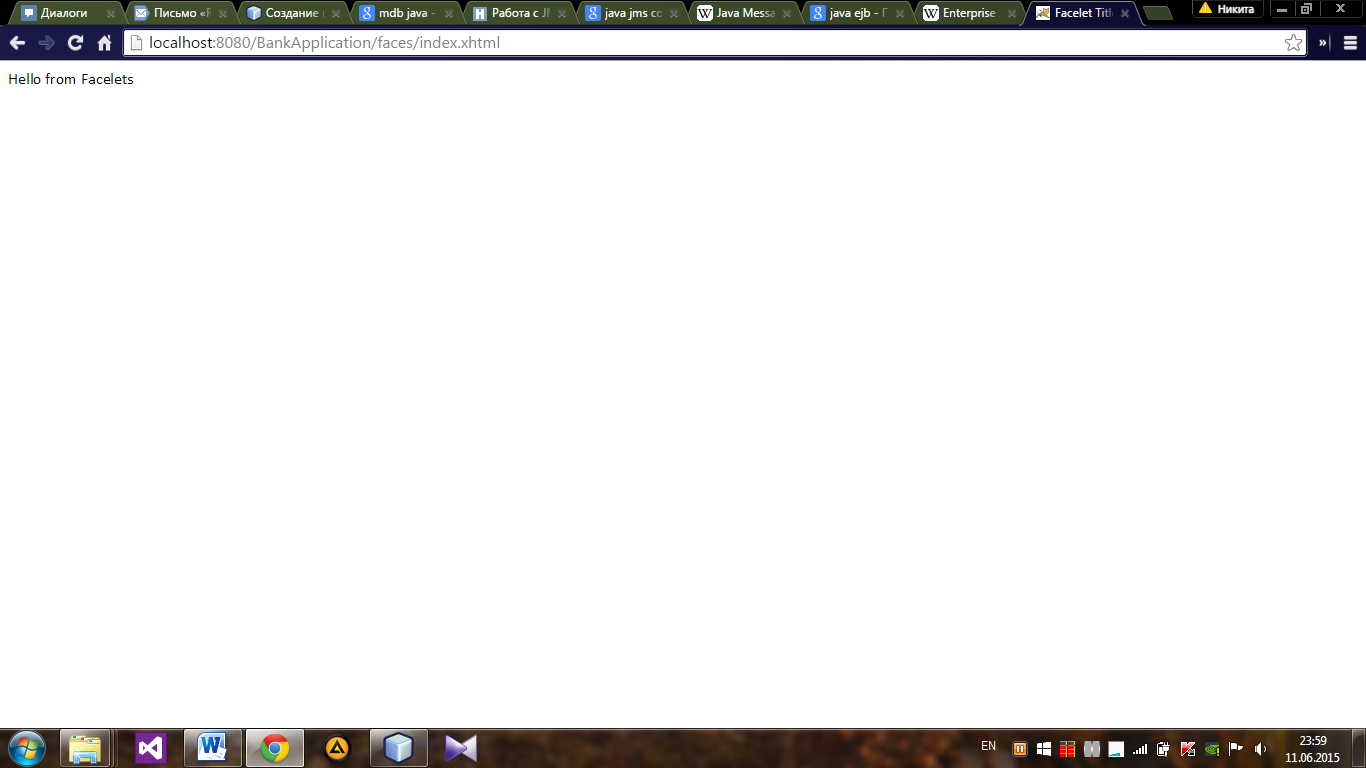


Рисунок 2 – Результат нажатия кнопки «Open account»

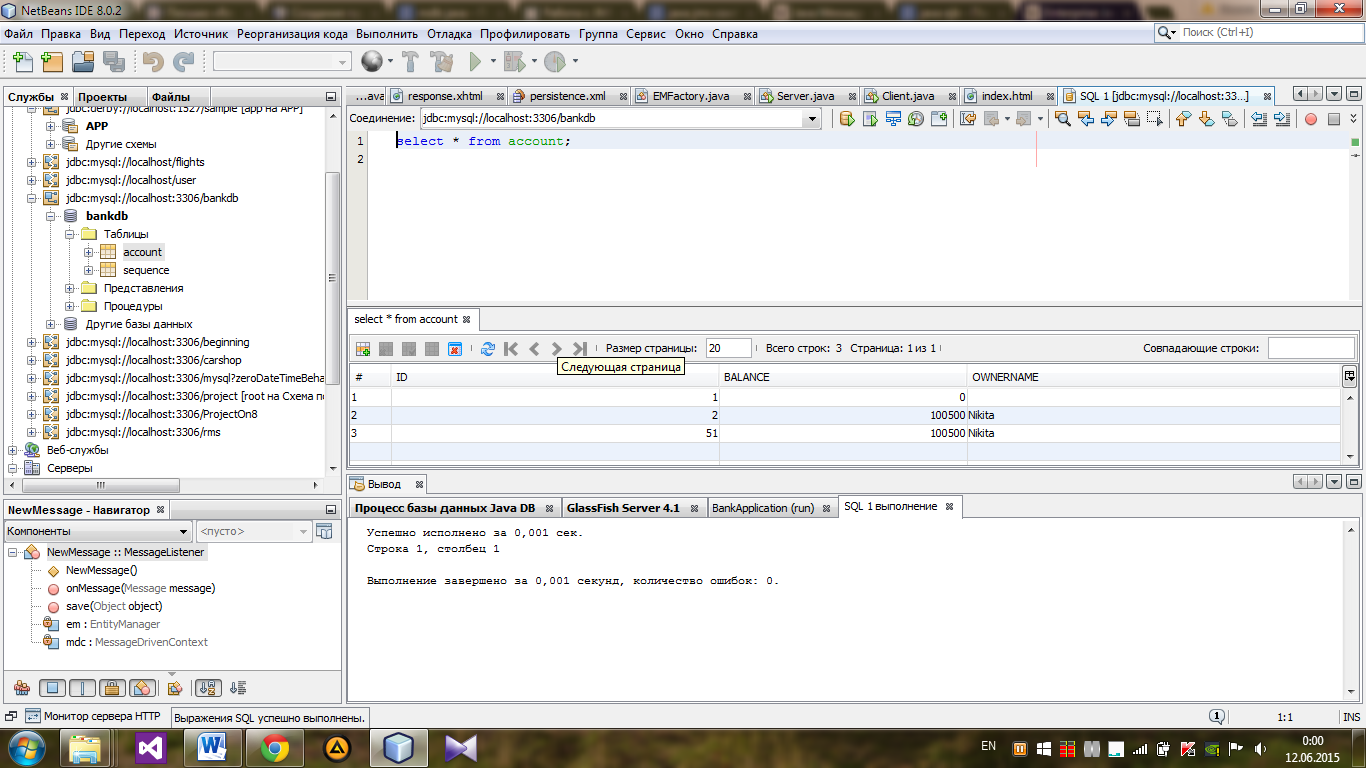


Рисунок 3 – Содержимое таблицы account базы bankdb

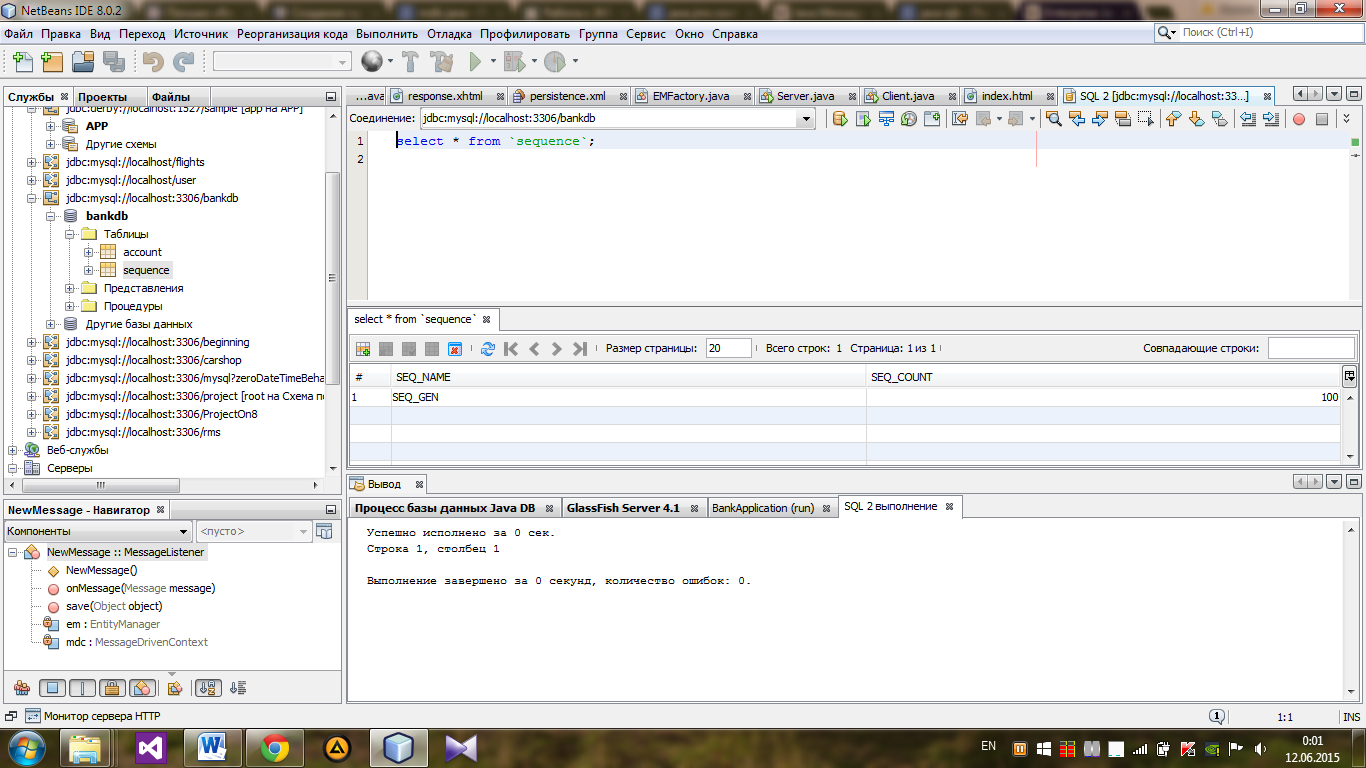


Рисунок 4 – Содержимое таблицы sequence базы bankdb

**Вопросы:**

1. Как сконфигурирован JPA?

JPA конфигурируется в файле persistence.xml. В нем содержится информация о сервере БД, его порте, пользователе и пароле, названии БД, провайдере персистентности и т.д. В данном случае, информация о соединении хранится на сервере glassfish, конкретно в пуле соединений. Через подключаемый в persistence.xml JDBC – ресурс bank, который также определен в glassfish, приложение получает всю необходимую информацию о соединении с БД. Информация о классах-сущностях хранится в аннотациях к классу Account (@Entity) и к классу-фасаду AccountFacade (@PersistenceContext).

1. Чем отличается JSF от JSP?

В отличии от JSP, управляемыми запросами, подход JSF основывается на использовании компонентов. Состояние компонентов пользовательского интерфейса сохраняется, когда пользователь запрашивает новую страницу и затем восстанавливается, если запрос повторяется. Также JSF это фреймворк, часто реализующийся на основе JSP. JSP используют сервлеты (и сами ими являются) для обмена сообщениями. JSF работает не с сообщениями, а с состояниями.

1. Что такое dependency injection?

Позволяет «вживлять» в бин другой бин в качестве управляющего свойства.

1. Что такое inversion of control?

Инверсия управления (МОК) - контроль и управление полезных серверных объектов в значительной степени передается в контейнер, а не остается на разработчике приложения. Иначе - это некий абстрактный принцип, набор рекомендаций для написания слабо связанного кода. Суть которого в том, что каждый компонент системы должен быть как можно более изолированным от других, не полагаясь в своей работе на детали конкретной реализации других компонентов.

1. Отличие stateless от statefull beans?

Stateless bean сохраняет свое содержание только на протяжении своего существования. После его уничтожения, информация, содержащаяся в нем, теряется. Statefull bean обладает персистентностью, т.е. хранит свои данные в БД, и поэтому может восстановить свою информацию после уничтожения и повторного создания.

Stateful может хранить какое-либо состояние между запросами, соответственно контейнер имеет пул таких бинов и для каждой сессии выделяет отдельный экземпляр. Stateless никакого состояния не хранит, а следовательно может использоваться один и тот же экземпляр.