Санкт-Петербургский государственный политехнический университет имени Петра Великого
Институт информационных технологий и управления
Кафедра компьютерных систем и программных технологий

Отчёт по курсовой работе. Вариант №13

Дисциплина: Распределенные вычисления и сети

Выполнил студент гр. 63501/2 _____ А.М. Зинченко
(подпись)
Руководители _____ И.В. Стручков
(подпись)

“_” _____2016 г.

Санкт - Петербург
2016

1. Задача

Реализовать электронную платежную систему. Операции удаленного объекта: создание счета с начальной суммой, пополнение счета, выполнение платежа с указанием получателя и цели, просмотр истории операций. Сериализуемый объект: платеж (получатель, цель, сумма). Пополнение счета считать особым видом платежа и также отображать в истории операций.

2. Модель предметной области

Модель предметной области строится в форме диаграммы классов UML, однако, важно отличать классы предметной области от классов, которые используются в программной реализации системы. Фактически, классы предметной области — это просто отражение понятий реального мира и их взаимосвязей. Они не имеют прямого отношения к программным классам.

Определим понятия предметной области:

- Платежная система
- Счет
- Платеж
- Цель платежа
- История платежей

Отразим отношения между ними (рис.1):

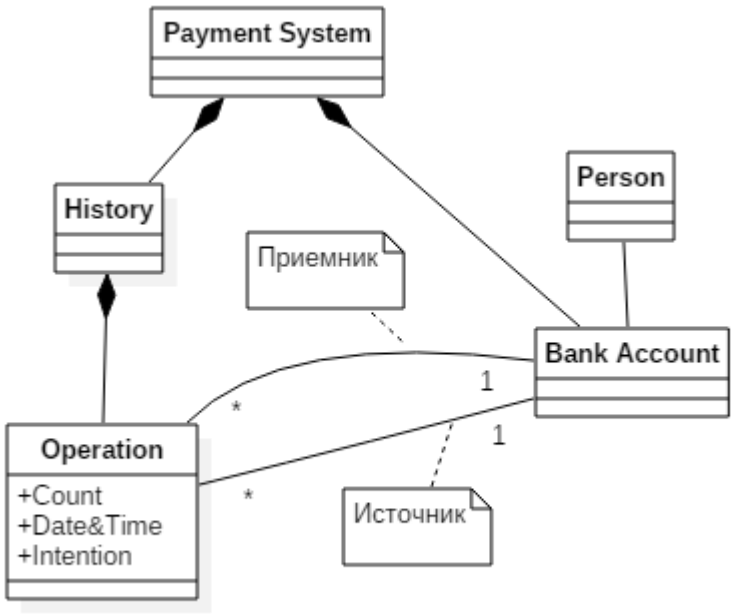


Рис.1.Модель предметной области

3. Варианты использования

Варианты использования описывают возможные действия пользователя и реакцию системы на эти действия. Используется текстовый способ описания:

Аутентификация пользователя

1. Пользователь вводит логин и пароль для входа в систему
2. Система проверяет правильность логина и пароля
3. Система выдает токен клиенту

Альтернатива: 3а. Логин и/или пароль неверны - выдать ошибку

Завершение работы

1. Клиент отсылает токен с командой завершения сессии
2. Система проверяет, что данный токен активен
3. Система удаляет информацию о связи токена с привязанным пользователем

Альтернатива: 3а. Токен неактивен - не производить никаких действий

Примечание: Перед каждым из дальнейших действий клиент должен отправить токен на проверку. Лишь в случае активности токена будет произведено действие, иначе - отказ.

Добавление пользователя

1. Пользователь вводит логин, пароль и начальную сумму
2. Система проверяет, что введенный логин не занят
3. Система добавляет пользователя с начальной суммой

Альтернатива: 3а. Логин занят - выдать ошибку

Выполнение платежа

1. Пользователь вводит логин пользователя, которому предназначается платеж, сумму и цель платежа
2. Система проверяет, что имеется такой пользователь, а также наличие средств на счету отправителя
3. Со счета отправителя списываются средства, на счет получателя зачисляются средства, и выполняется запись об операции в историю

Альтернатива: 3а. Нет такого пользователя или недостаточно средств - выдать ошибку

Пополнение счета

1. Пользователь вводит сумму
2. На счет данного пользователя зачисляется соответствующая сумма, выполняется запись в историю операций

Отображение истории операций

1. Пользователь может ввести информацию о периоде, слова, фигурирующие в цели или логин получателя

2. Система выдает информацию об операциях с участием данного пользователя с введенными фильтрами

4. Диаграмма классов проектирования

Эта диаграмма классов внешне похожа на модель предметной области, но существенно отличается от нее по сути. Элементами данной модели являются уже не абстрактные понятия, а реальные классы и интерфейсы будущей программной системы.

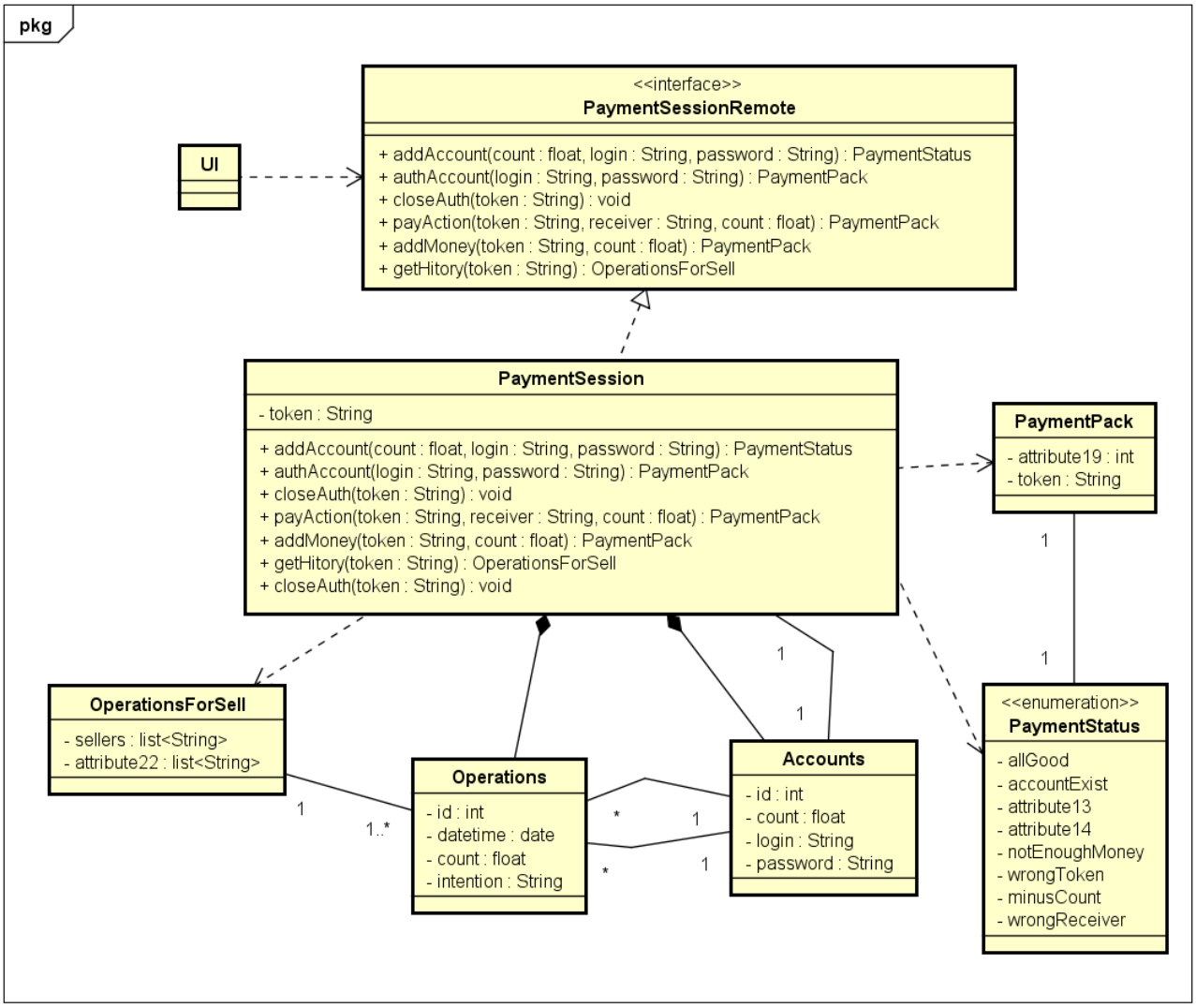


Рис.2.Диаграмма классов проектирования

5. Диаграмма последовательностей

При создании диаграммы классов проектирования также строится диаграмма последовательностей, отражающая взаимодействие объектов на данной диаграмме и конкретизирующее варианты использования. Приведем некоторые из них:

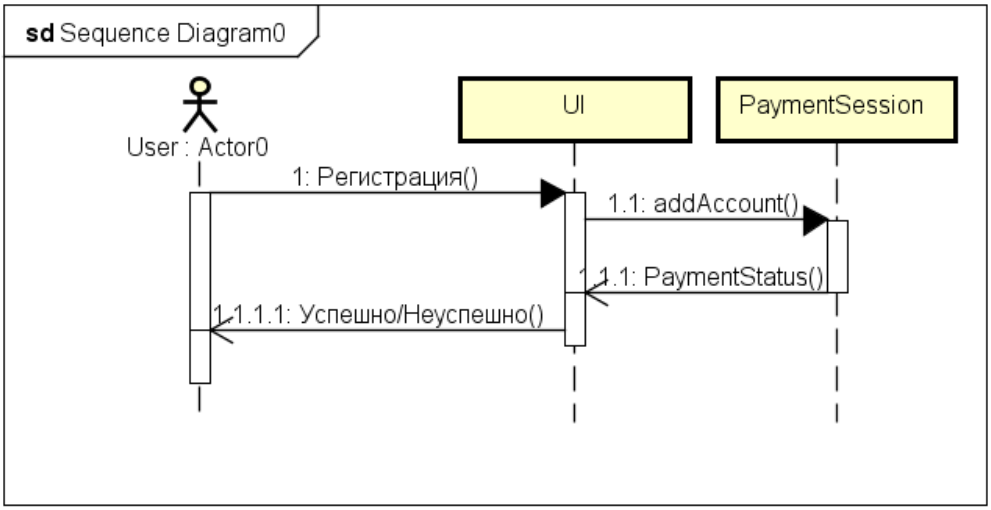


Рис.3.Регистрация нового пользователя

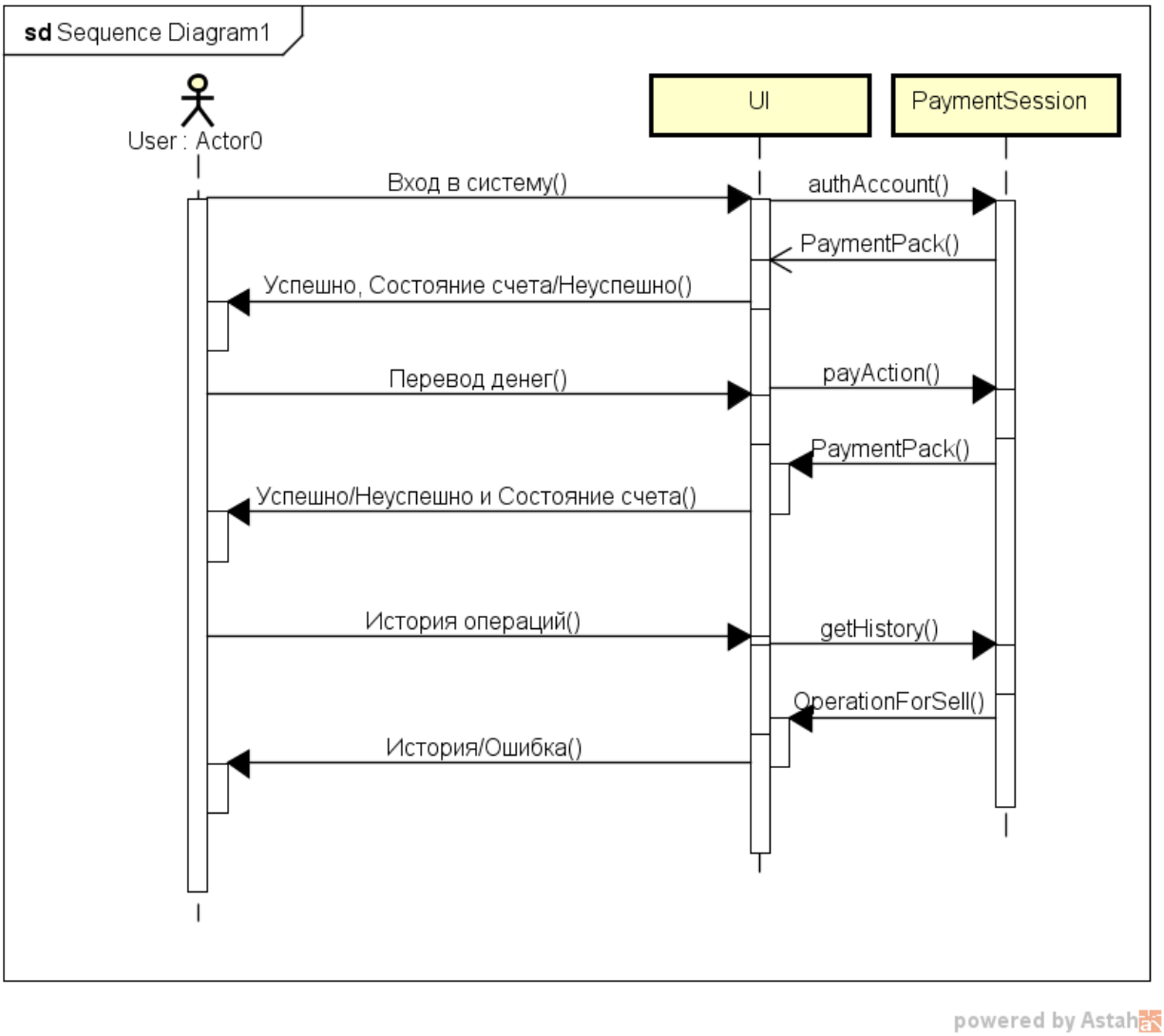


Рис.4. Пример использования системы

6. Описание классов

Классы «Accounts» и «Operations» являются сущностными классами, изменения в которых приводят к изменениям в соответствующих таблицах БД.

Перечисление «PaymentStatus» является частью возвращаемого значения практически для всех методов и служит для индицирования успешного или неуспешного выполнения операции с указанием причины

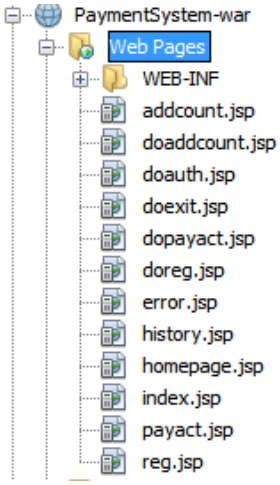
Класс «PaymentPack» служит для одновременной передачи данных, например, при авторизации необходимо передать счет, токен, а также индицировать успешный/неуспешный вход в систему (с помощью поля типа PaymentStatus).

Основной класс – PaymentSession. Для взаимодействия с клиентом используется интерфейс PaymentSessionRemote. Класс объявлен как Stateful, поскольку при успешном входе в полях класса хранится токен и указатель на рабочий аккаунт. Кратко опишем данные методы:

- public PaymentStatus addAccount(float count, String login, String password)
  Добавление нового пользователя
- public PaymentPack authAccount(String login, String password)
  Аутентификация пользователя
- public void closeAuth(String token)
  Выход из системы удаление SessionBean (функция с аннотацией @Remove)
- public PaymentPack payAction(String token, String receiver, float count)
  Перевод средств другому пользователю
- public PaymentPack addMoney(String token, float count)
  Пополнение счета
- public OperationsForSell getHitory(String token)
  Получение истории операций
- private void checkToken(String token) throws java.lang.Exception
  Проверка токена

7. Описание слоя представления

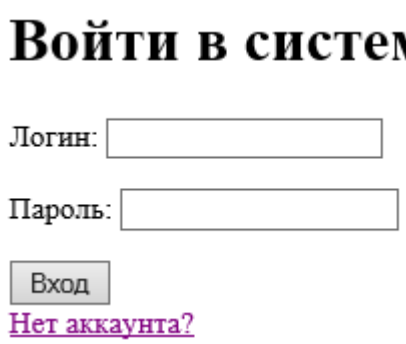
Клиентская часть данного распределенного приложения реализована с помощью jsp-страниц. Приведем их перечень (демонстрация скриншотов и описание даны в пункте Тестирование):



8. Тестирование и демонстрация работы

8.1. Вход в систему

Реализован с помощью страниц index.jsp и doauth.jsp.

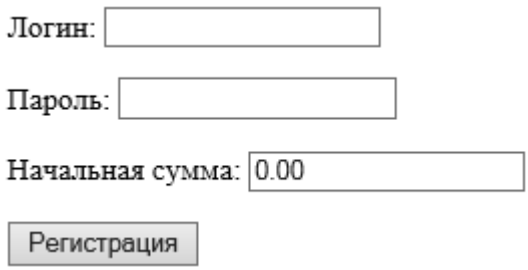


Ответы системы при альтернативном использовании:

Неверный пароль
wrongPassword
Неверный логин
accountNotExist

8.2. Добавление пользователя

Реализовано с помощью страниц reg.jsp и doreg.jsp.



Ответы системы при альтернативном использовании:

Существующий аккаунт
accountExist
Отрицательная сумма
minusCount

8.3. Домашняя страница

Реализовано с помощью страниц homepage.jsp. Отображает имя пользователя и остаток средств.

Приветствую, zkoalex!

Средств: 180,00

Пополнить счет
Выполнить перевод
История операций
Выход

8.4. Страница пополнения счета

Реализовано с помощью страниц addcount.jsp и doaddcount.jsp

Внесено: 0.00

Внести

Ответы системы при альтернативном использовании:

Отрицательный ввод
minusCount

8.5.Страница перевода средств

Реализовано с помощью страниц payact.jsp и dopayact.jsp

Получатель:

Сумма: 0.00

Перевести

Ответы системы при альтернативном использовании:

Несуществующий аккаунт
accountNotExist
Отрицательная сумма
minusCount

8.6. История операций

Реализовано с помощью страницы history.jsp

Дата/Время	Отправитель	Получатель	Сумма	Цель
Fri Nov 25 15:12:02 MSK 2016	zkoalex	zkoalex	3.0	null
Fri Nov 25 15:17:13 MSK 2016	zkoalex	fd	5.0	null
Fri Nov 25 15:23:16 MSK 2016	fd	zkoalex	100.0	null
Thu Dec 15 00:57:06 MSK 2016	zkoalex	zkoalex	5.0	null

В данном списке присутствует операция пополнений счетов и переводов, что свидетельствует о правильной работе системы при корректном использовании.

8.7. Выход из системы

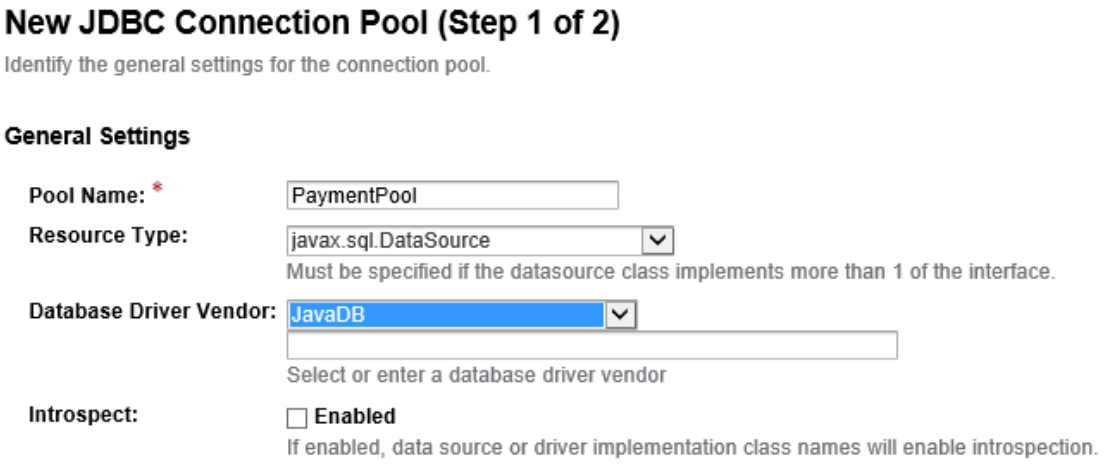
Реализован с помощью страницы doexit.jsp. В случае успешного выхода перенаправляет на главную страницу (index.jsp)

8.8. Страница ошибок

Реализовано с помощью страницы error.jsp Печать возвращаемой системой переменной типа PaymentStatus.

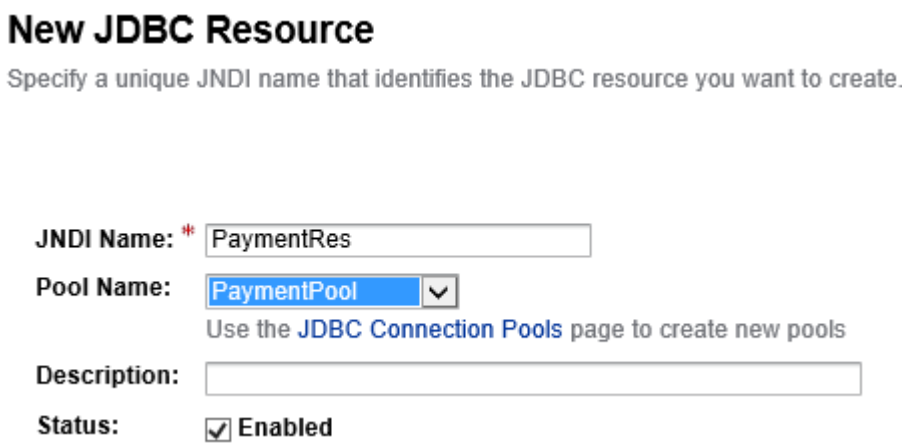
9. Инструкция системного администратора

1. Скачать и установить Java EE SDK
2. В переменные окружения добавить JAVA_HOME
   C:\Program Files\Java\jdk1.8.0_60
3. Скачать и установить ParayaServer
4. В папку lib используемого домена скопировать RemoteNeeds.jar
5. В папку glassfish\databases скопировать папку PaymentDB
6. Открыть окно команд и перейти в папку payara41\bin
7. Выполнить команду asadmin --port 27029
8. Выполнить команду start-database
9. Выполнить команду start-domain <Имя домена>
10. Открыть браузер для конфигурации домена http://localhost:27029/
11. Создать JDBC Connection Pool



Задать следующие параметры

12. Создать JDBC resource



13. Развернуть приложение с помощью команды deploy --force=true <Путь>PaymentSystem.ear
14. Открыть браузер на странице http://localhost:27061/PaymentSystem-war/

10. Инструкция пользователя

При наличии аккаунта введи логин и пароль и войдите в систему, при отсутствии – необходимо пройти простую регистрацию, введя свой логин, пароль и начальную сумму. Обратите внимание, что введенный логин может быть занят, поэтому потребуется ввести новый логин.

После входа в систему выберите необходимый пункт:

1. Пополнение счета – введите сумму для пополнения (не допускаются отрицательные значения)
2. Перевод другому пользователю – введите существующий логин пользователя и сумму перевода (на счете должно быть достаточно средств)
3. Просмотр истории операций
4. Выход из системы – по завершении работы не забудьте выполнить выход из системы

11. Вывод

В данной работе были изучены принципы работы с одним из существующих ORM, принципы работы технологии EJB, реализовано клиент-серверное приложение для электронных платежей.

Развивать приложение можно по следующим направлениям:

- доработка дизайна клиента и добавление новых функций (например, фильтрация истории операций),
- изучение и использование методов аутентификации, предлагаемых Java,

Стоит отметить, что использование технологий, предлагаемых Java, значительно ускоряет и упрощает разработку распределенных приложений.