**Лабораторная работа № 9**

**«Основы разработки Web-сервисов»**

**Цель работы:** изучить основы разработки Web-сервисов с использованием инструментальной среды NetBeans.

**Краткие теоретические сведения**

***Основы Web-сервисов***

Web-сервисы преобразуют XML-документы (Extensible Markup Language, XML) в ИТ-системах. Web-сервисы - это XML-приложения, осуществляющие связывание данных с программами, объектами, базами данных либо с деловыми операциями целиком. Между Web-сервисом и программой осуществляется обмен XML-документами, оформленными в виде сообщений. Стандарты Web-сервисов определяют формат таких сообщений, интерфейс, которому передается сообщение, правила привязки содержания сообщения к реализующему сервис приложению и обратно, а также механизмы публикации и поиска интерфейсов.

Web-сервисы могут использоваться во многих приложениях. Независимо от того, откуда запускаются Web-сервисы, с настольных компьютеров клиентов или с переносных, они могут использоваться для обращения к таким интернет-

приложениям, как система предварительных заказов или контроля выполнения заказов. Web-сервисы пригодны для В2В-интеграции (business-to-business), замыкая приложения, выполняемые различными организациями, в один производственный процесс. Web-сервисы также могут решать более широкую проблему интеграции приложений предприятия (Enterprise Application Integration, EAI), осуществляя связь нескольких приложений одного предприятия с несколькими другими приложениями, размещенными как "до", так и "после" брандмауэра. Во всех перечисленных случаях технологии Web-сервисов являются "связующим звеном", объединяющим различные части программного обеспечения.

Как видно из рисунка, Web-сервисы представляют собой оболочку, обеспечивающую стандартный способ взаимодействия с прикладными программными средами, такими как системы управления базами данных (СУБД), .NET, J2EE (Java2 Platform, Enterprise Edition), CORBA (Common Object Request Broker Architecture), посредники пакетов планирования ресурсов предприятия (Enterprise Resource Planning, ERP), брокеров интеграции и пр.



Рисунок – Web-сервисы взаимодействуют с прикладными системами

Интерфейсы Web-сервисов получают из сетевой среды стандартные XML-сообщения, преобразуют XML-данные в формат, "понимаемый" конкретной прикладной программной системой, и отправляют ответное сообщение (последнее - не обязательно). Программная реализация Web-сервисов (базовое программное обеспечение, нижний уровень) может быть создана на любом языке программирования с использованием любой операционной системы и любого связующего программного обеспечения (middleware).

Web-сервисы объединяют программирование и концепции Сети. Web-сервисы сочетают параметры программных приложений и абстрактные характеристики Сети. Современные интернет-технологии частично достигают своих целей, поскольку они определены на очень высоком отвлеченном уровне, что обеспечивает совместимость с любой операционной системой, любым программным и аппаратным обеспечением. Инфраструктура, основанная на применении Web-сервисов, пользуется этим уровнем абстракции и включает в себя связанную с данными семантическую информацию, то есть Web-сервисы определяют не только данные, но и порядок обработки и преобразования этих данных в базовые программные приложения и обратно.

***Основные положения модели Web-сервисов***

Концепция Web-сервисов реализуется при помощи ряда технологий, которые стандартизованы World Wide Web Consortium (W3C).

Взаимосвязь этих технологий представлена на рисунке.

Рисунок – Взаимосвязь технологий Web-сервисов

Web-сервисы являются одним из вариантов реализации компонентной архитектуры. XML является фундаментом для создания большинства технологий, связанных с Web-сервисами.

Для удаленного взаимодействия с Web-сервисами используется Simple Object Access Protocol (SOAP). SOAP обеспечивает взаимодействие распределенных систем, независимо от объектной модели, операционной системы или языка программирования. Данные передаются в виде особых XML документов особого формата.

Согласно определению W3C, Web-сервисы - это приложения, которыедоступны по протоколам, которые являются стандартными для Интернет. Нет требования, чтобы Web-сервисы использовали какой-то определенный транспортный протокол. Спецификация SOAP определяет, каким образом связываются сообщения SOAP и транспортный протокол.

Сообщение SOAP является обычным XMl-документом. Этот документ состоит из следующих XML-элементов:

* SOAP envelope, который определяет содержание послания
* SOAP header, в нем заключается заголовочная информация (необязательный элемент)
* SOAP body, в нем содержится информация вызовов и ответов на вызовы

**Пример SOAP-запроса**

<soap: Envelope>

<soap: Body>

<GetPrice>

<Item>Apples</Item>

</GetPrice>

</soap: Body>

</soap: Envelope>

Наиболее часто реализуется передача SOAP сообщений по протоколу HTTP. Также широко распространено использование в качестве транспортного протокола SMTP, FTP, TCP.

Согласно определению W3C, "WSDL - формат XML для описания сетевых сервисов как набора конечных операций, работающих при помощи сообщений, содержащих документно-ориентированную или процедурно-ориентированную информацию". Документ WSDL полностью описывает интерфейс Web-сервиса с внешним миром. Он предоставляет информацию об услугах, которые можно получить, воспользовавшись методами сервиса, и способах обращения к этим методам.

Технология Universal Description, Discovery and Integration (UDDI) предполагает ведения реестра Web-сервисов. Подключившись к этому реестру, потребитель сможет найти Web-сервисы, которые наилучшим образом удовлетворяют его потребностям. Технология UDDI дает возможность поиска и публикации нужного сервиса, как человеком, так и программой-клиентом. Поиск и публикация в реестре предоставляется программе-клиенту как набор Web-сервисов реестра UDDI.

Web-сервисы позиционируются как программное обеспечение промежуточного слоя. Использовать Web-сервисы могут как клиентские приложения, непосредственно работающие с пользователем, так и другие приложения (в том числе и другие Web-сервисы).

Концепция Web-сервисов включает в себя возможность ведения реестра Web-сервисов. Описание интерфейса может быть получено из такого реестра. После создания и внедрения нового Web-сервиса, имеет смысл зарегистрировать его в реестре. Тогда клиенты при поиске сервисов, реализующих исходный интерфейс, получат указание и на новый Web-сервис.

Использование Web-сервиса как строительного блока при создании приложения. Приложение может использовать Web-сервисы как удаленные компоненты, которые предоставляют определенную функциональность. Существуют различные сервисы, которые предоставляют качественное решение таких задач как аутентификация, ведение календаря, отправка сообщений, поиск, перевод и т. п.



Рисунок - Использование интерфейсов существующих Web-сервисов

Кроме этого, и возможны другие варианты использования Web-сервисов. Например, существуют исследования по использованию Web-сервисов для построения распределенных вычислительных и информационных систем и одноранговых и со сложной иерархической структурой.

***Взаимодействие Web-сервисов***

Web-сервисы поддерживают несколько парадигм обмена сообщениями. Уровень абстракции, на котором оперируют Web-сервисы, подразумевает такие стили взаимодействия, как эмуляцию удаленного вызова процедуры (Remote Procedure Call, RPC), асинхронный обмен сообщениями, однонаправленную передачу сообщений, широковещание и публикацию/подписку. Основные СУБД, такие как Oracle, SQL Server и DB2, поддерживают анализ XML и службы преобразования, обеспечивая непосредственное взаимодействие между Web-сервисами и СУБД. Производители связующего программного обеспечения обычно также предоставляют возможность привязки Web-сервисов к своим программным системам (серверам приложений и брокерам интеграции). Следовательно, для пользователя взаимодействие с Web-сервисами может проявляться в интерактивной или пакетной форме, поддерживающей синхронную и асинхронную модели связи; а также как пользовательский интерфейс, написанный с использованием Java, C#, офисных приложений, браузеров или "толстых" клиентов СУБД. Такое взаимодействие может привязываться к любому типу базовой (более низкого уровня) программной системы.

Web-сервисы выполняют RPC- и документно-ориентированное взаимо-действия. Стандарты и технологии Web-сервисов обычно подразумевают два основных типа моделей взаимодействия приложений:

* удаленный вызов процедуры (онлайновая)
* документно-ориентированный (пакетная)

Эти два типа взаимодействия рассматриваются в последующих разделах.

***RPC-ориентированные взаимодействия***

RPC-ориентированные взаимодействия удобны для краткого обмена данными. В RPC-ориентированном взаимодействии запросы Web-сервисов приобретают форму вызова метода или процедуры с соответствующими входными или выходными параметрами. В отличие от документно-ориентированного взаимодействия, RPC-ориентированное взаимодействие производит отправку документа, специально отформатированного для передачи в отдельную логическую программу или базу данных.

***Документно-ориентированное взаимодействие***

Документно-ориентированные взаимодействия удобны для обмена большими объемами данных. При документно-ориентированном взаимодействии запросы Web-сервиса имеют форму завершенного XML-документа, предназначенного для обработки целиком.

***Технология Web-сервисов***

Порядок описания, поиска и взаимодействия Web-сервисов друг с другом определяют стандарты. Взаимодействующие через Интернет программы должны уметь обнаруживать друг друга, находить информацию, позволяющую им осуществить связь, понимать, какая модель контактирования должна быть применена (простая, типа "запрос/ответ", или более сложная последовательность), и договариваться об использовании таких услуг, как защита информации, подтверждение передачи сообщений и составление сделок. Некоторые из этих сервисов реализуются существующими технологиями и предлагаемыми стандартами, а другие - нет. Использующее Web-сервисы сообщество стремится удовлетворить все эти требования, но это - эволюционный процесс, как и сам Интернет. С самого начала инфраструктура и стандарты Web-сервисов подразумевали возможность расширения (так же как до них XML и HTML), что позволяет использовать их сразу же после появления новых стандартов и технологий.

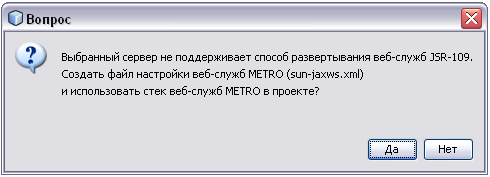
Для транспортировки и преобразования данных в программы и обратно Web-сервисы требуют использования нескольких смежных XML-технологий.

* Язык XML (Extensible Markup Language) - фундамент, на котором строятся Web-сервисы. Он предоставляет язык определения данных и порядок их обработки. XML представляет семейство связанных спецификаций, публикуемых и поддерживаемых интернет-консорциумом (World Wide Web Consortium, W3C) и другими организациями.
* WSDL (Web Services Description Language) - технология, основанная на XML, определяющая интерфейсы Web-сервисов, типы данных и сообщений, а также модели взаимодействия и протоколы связывания.

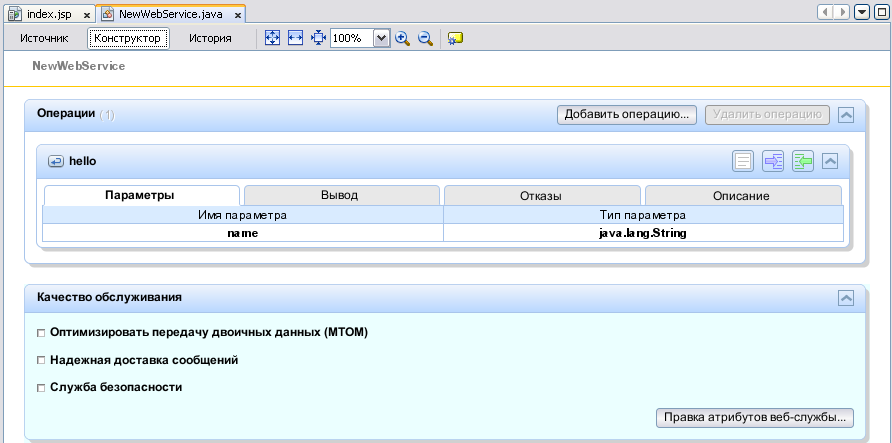
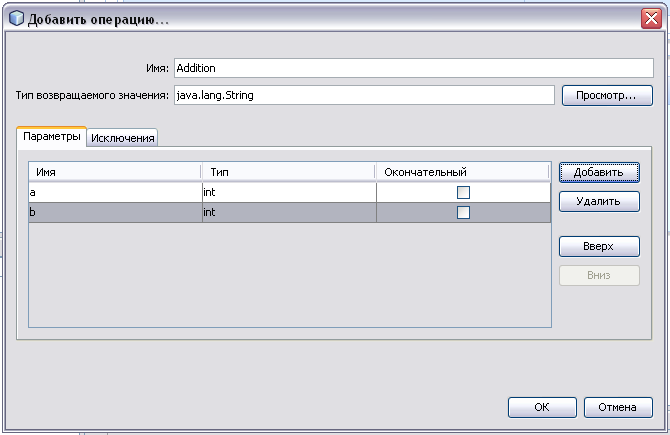
***Создание проекта в среде NetBeans***

Здесь рассматривается пример создания простого Web-сервиса в среде NetBeans, который без особых усилий будет развернут на Web-сервере Tomcat:

1. Выберите «Файл» > «Создать проект». Выберите "Веб-приложение" из категории "Java Web". Определите имя (например, SimpleWebService) и расположение проекта. Далее выбирается сервер (в данном случае Tomcat). Этот шаг завершается нажатием «Далее» и «Готово».
2. Для создания простого Web-сервиса необходимо выполнить следующие действия:

* нажатием правой кнопкой на названии проекта выбирается «Новый/Другое/Веб службы/Веб служба»
* вводится название проекта (например, Вводим название проекта NewWebService) и нажимается кнопка «Готово». Если появится сообщение вида,

Нажмите кнопку «Да».

* Вверху выбирается вкладку «Конструктор», нажимается «Добавить операцию…» — чтобы добавить операцию, которую будет выполнять сервис
* Далее добавляется описание операции:
* После этого на вкладке «Источник» пишется небольшой код:

/\*\*

\* Операция веб-службы

\*/

@WebMethod(operationName = "Addition")

public int Addition(@WebParam(name = "a") int a, @WebParam(name = "b") int b)

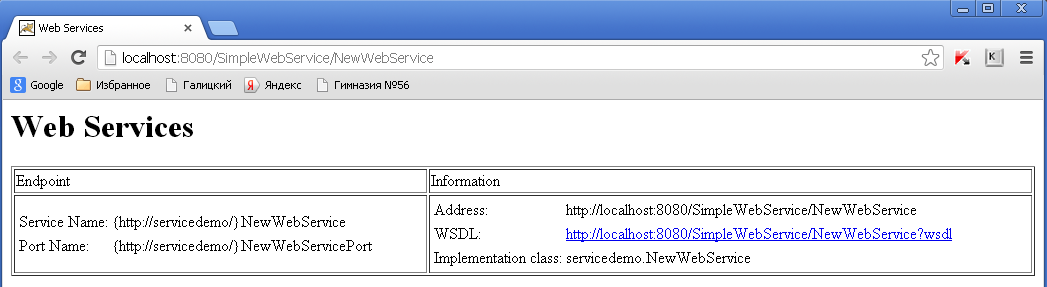
{

int c = a + b;

return c;

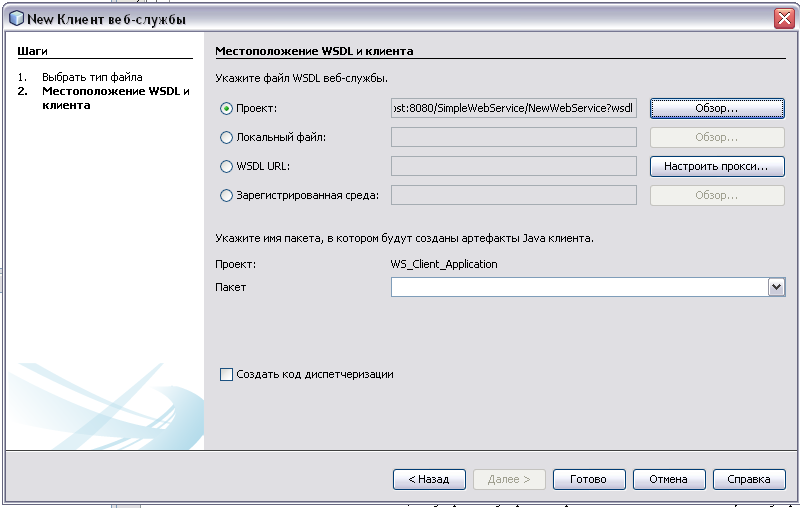
}

1. Для развертывания Web-сервиса нажимается правая кнопка на проекте и выбирается «Развернуть». Для проверки Web-сервиса необходимо на узле «Веб-службы» проекта правой кнопкой выбрать «Тестировать веб-службу». Для Web-сервера Tomcat на экране появится

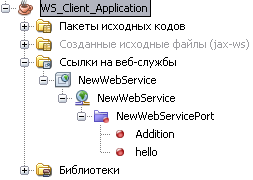


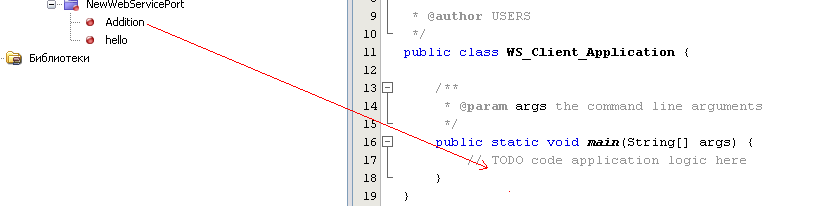
1. После успешного развертывания веб-службы необходимо создать клиент для использования метода веб-службы Addition, например, выполнить последовательность действий по созданию консольного приложения:

* Выбрать «Файл» > «Создать проект», «Приложение Java» в категории «Java». Присвойте проекту имя WS\_Client\_Application. Не снимайте флажок «Создать главный класс» и оставьте все прочие значения по умолчанию. Нажмите кнопку «Готово».
* Щелкните правой кнопкой мыши узел CalculatorWS\_Client\_Application и выберите «Создать > Новое>Другое…>Веб-службы>Клиент веб-службы». Нажать кнопку «Далее». Появится мастер создания клиента веб-службы.
* Выберите проект в виде исходного файла WSDL. Нажмите кнопку «Обзор». В раннее созданном проекте SimpleWebService перейдите к веб-службе NewWebService. Выберите веб-службу и нажмите кнопку «ОК».
* Не выбирайте имя пакета. Оставьте это поле пустым. Оставьте значения остальных параметров по умолчанию и нажмите кнопку «Готово».



* В окне «Проекты» появится новый клиент веб-службы с узлом для созданного метода Addition:



* Дважды щелкните главный класс, чтобы открыть его в редакторе исходного кода. Перетащите узел Addition в местоположение ниже метода main():

На экране должен появиться следующий код:

public static void main(String[] args)

{

// TODO code application logic here

}

private static int addition(int a, int b)

{

servicedemo.NewWebService\_Service service = new servicedemo.NewWebService\_Service();

servicedemo.NewWebService port = service.getNewWebServicePort();

return port.addition(a, b);

}

**Примечание**. В качестве альтернативы, вместо перетаскивания узла Addition можно щелкнуть правой кнопкой мыши в редакторе, а затем выбрать «Вставка кода» > «Вызов операции веб-службы…».

* В теле метода main() замените комментарий TODO кодом, который инициализирует значения для i и j, вызывает addition() и распечатывает результат

public static void main(String[] args)

{

try

{

int i = 3;

int j = 4;

int result = addition(i, j);

System.out.println("Result = " + result);

} catch (Exception ex)

{

System.out.println("Exception: " + ex);

}

}

* По правой кнопке мыши на узле проекта выбрать «Выполнение».

**Порядок выполнения работы**

1. Ознакомиться с теоретическими сведениями
2. Разработать структуру веб-сервиса согласно варианту задания
3. Создать веб-сервис
4. Создать клиент веб-сервиса
5. Составить отчет о выполнении лабораторной работы

|  |  |
| --- | --- |
| Вариант | Web-сервис |
| 1 | Температурные шкалы Цельсия, Фаренгейта, Кельвина, Реомюра |
| 2 | Метрические и британские единицы массы |
| 3 | Метрические и британские единицы длины |
| 4 | Метрические и британские единицы объема |
| 5 | Единицы угловой меры |
| 6 | Морские единицы длины и расстояния |
| 7 | Старорусские единицы длины и расстояния |
| 8 | Старорусская мера жидкости |

**Содержимое отчета**

1. Название и тема лабораторной работы
2. Цель лабораторной работы
3. Краткие теоретические сведения
4. Ход выполнения работы
5. Тексты программ, конфигурационные файлы, используемые для выполнения лабораторной работы
6. Выводы

**Контрольные вопросы**

1. Что такое веб-сервис?
2. Что такое SOAP?
3. Структура SOAP-пакета
4. Назовите два основных типа моделей взаимодействия приложений технологии веб-сервисов
5. Какие технологии используют веб-сервисы для транспортировки и преобразования данных в программы и обратно?
6. Что такое WSDL?
7. Что такое RPC-ориентированные взаимодействия?
8. Способы взаимодействия веб-сервисов