

Кроссплатформенные программные системы

Методические указания к практическим занятиям

Составитель Тимофеев А.А.

Работа № 1. Определение структуры XML-документа средствами DTD и XML-схем

Данная работа предусматривает разработку формального описания структуры XML-документов, представляющих информацию из какой-то предметной области. Описание необходимо выполнить средствами DTD и в виде XML-схемы, при этом привести пример XML-документов, соответствующих DTD-описанию и схеме.

Технические требования для DTD-описания:

- 1) должно быть выполнено в отдельном внешнем файле, а не непосредственно в рамках документа-примера
- 2) документ-пример должен содержать ссылку на DTD-описание
- 3) DTD-описание должно демонстрировать использование конструкций описания элементов с вложенными элементами и применением следования, альтернативы, необязательности и повторяемости этих вложенных элементов
- 4) DTD-описание должно демонстрировать использование конструкций описания атрибутов, как обязательных, так и необязательных, в том числе со значением по умолчанию для атрибутов следующих типов: ID, IDREF или IDREFS, CDATA, а также одного из перечисляемых типов
- 5) DTD-описание должно демонстрировать использование параметризованных сущностей и ссылок на них
- 6) DTD-описание должно демонстрировать использование общих сущностей и ссылок на них в документе-примере
- 7) DTD-описание должно демонстрировать использование внешних неанализируемых сущностей в сочетании с описанием атрибута

Технические требования для XML-схемы:

- 1) схеме должно быть сопоставлено определенное пространство имен, при этом должна быть предусмотрена квалификация элементов, определенных схемой, в документе, построенном на ее основе
- 2) документ-пример должен содержать ссылку на XML-схему
- 3) схема должна демонстрировать использование конструкций для описания сложного типа, образованного вложенными элементами и атрибутами; при этом должен быть хотя бы один сложный тип, хотя бы один вложенный элемент которого был бы также сложного

типа; при описании сложных типов необходимо продемонстрировать использование не менее двух видов модельных групп (sequence, choice и all)

4) схема должна содержать описание абстрактного сложного типа и элемента этого типа, а пример документа - демонстрировать использование последнего

5) схема должна демонстрировать возможность описания нового сложного типа на базе как простого типа, так и сложного как путем расширения, так и путем ограничения

6) схема должна демонстрировать возможности описания новых простых типов на базе существующих (в том числе описание перечислимого типа) с использованием не менее трех видов фасетов

7) схема должна содержать описание ограничения уникальности, ключа и ссылки на ключ, применение чего должно иметь смысл в предметной области

Работа № 2. Основы JAXP. Программный анализ XML-документов

Данная работа предусматривает разработку программ(ы) анализа XML-документов, составленных в соответствии со схемой, разработанной в ЛР1. Суть анализа должна заключаться в формировании HTML-представления XML-документа, поданного программе на вход. Возможные способы реализации: 1) две программы с идентичным поведением, использующие средства DOM API и SAX API, соответственно; 2) одна программа, использующая средства StAX API.

Технические требования для программ:

- 1) предметная область должна предусматривать наличие данных трех типов: строкового, числового и типа дата
- 2) структура и наполнение XML-документа должны быть такими, чтобы часть данных можно было представить в табличной форме, и при этом в каждой строке таблицы было как минимум два числовых поля и как минимум одно поле типа дата
- 3) результат формирования HTML-представления должен содержать в табличной части вычисляемое поле, формируемое на основе данных строки; вычисление может быть любым, но осмысленным в терминах предметной области
- 4) результат формирования HTML-представления должен содержать в табличной части подвал с как минимум одним итогом; итоги могут вычисляться любым способом, но осмысленно в терминах предметной области
- 5) ввод каждая из программ должна брать со стандартного ввода, результат преобразования передавать на стандартный вывод
- 6) каждая из программ должна обеспечить обработку ошибок анализа документа соответствующими средствами, при этом в случае ошибки уровня `error` и `fatal error` анализ должен быть прекращен без выдачи результата преобразования на стандартный вывод, а при возникновении ошибки уровня `warning` анализ должен быть продолжен, а преобразование - выполнено; в любом случае при возникновении ошибки любого уровня необходимо на стандартный вывод об ошибках выдать информацию об уровне ошибки, системное сообщение об ошибке и место возникновения ошибки с указанием строки и столбца документа
- 7) при анализе документа необходимо выполнить проверку соответствия документа схеме, при этом предполагается, что файл со схемой поставляется вместе с программой и должен быть расположен в каталоге программы, из которого она и запускается, а анализируемый файл может и вовсе не содержать ссылки на схему

Работа № 3. Основы XSLT. Преобразования средствами JAXP

Данная работа предусматривает разработку описания XSLT-преобразования, результат которого был бы эквивалентен результату, получаемому в ЛР2. При этом собственно XSLT-преобразование необходимо выполнить программно, используя средства JAXP, для программы сохраняются все условия, которые указаны в задании для предыдущей работы, но добавляются те, что указаны ниже.

Технические требования для XSLT-описания и программы преобразования:

- 1) XSLT-преобразование должно демонстрировать использование именованных параметризованных шаблонов
- 2) XSLT-преобразование должно демонстрировать использование инструкций ветвления и цикла
- 3) XSLT-преобразование должно демонстрировать использование инструкций нумерации и сортировки
- 4) XSLT-преобразование должно содержать указание метода вывода - HTML, сформированное с учетом требований спецификации HTML v4.01
- 5) программа преобразования должна использовать источник и результат преобразования различных типов (например, SAXSource и DOMResult)
- 6) программа преобразования должна обеспечить обработку ошибок этапа преобразования в дополнение к обработке ошибок этапа анализа по такому же принципу

Работа № 4. Основы JAXB

Данная работа предусматривает разработку программы, использующей JAXB для обработки XML-документа. Обработка должна заключаться в формировании HTML-представления XML-документа, поданного на вход программы. Кроме того, в процессе обработки должно изменяться содержимое либо структура полученного при анализе графа объектов, и должна быть предусмотрена возможность сохранения измененного графа объектов в виде XML-документа.

Технические требования для программы совпадают с требованиями, указанными в ЛР2.

Работа №5. Разработка RESTсервиса

Задание

1. Разработать RESTful веб-сервис любым из доступных способов: в виде сервлета, в виде провайдера JAX-WS, в виде ресурса JAX-RS, при помощи Spring MVC.

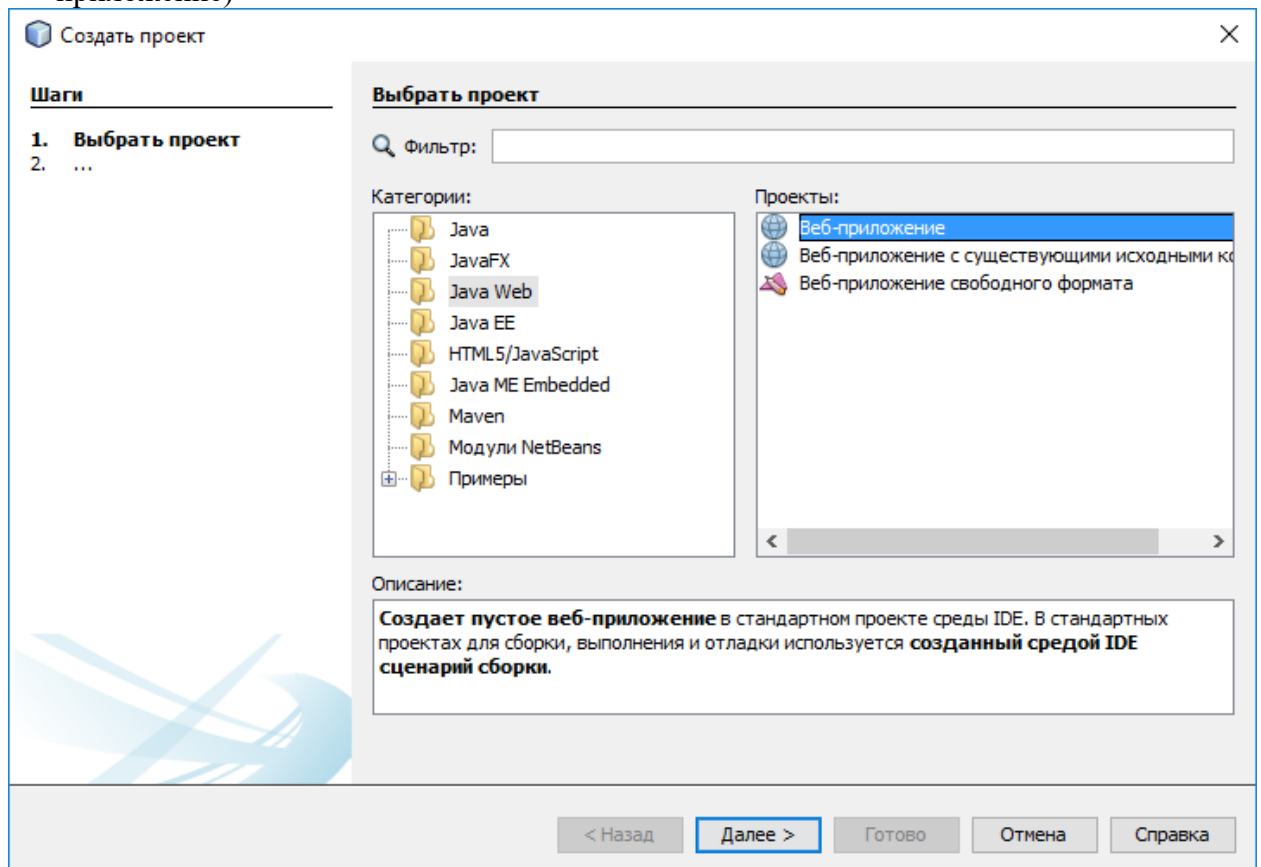
Средства выполнения

1. NetBeans IDE 8.2
2. Glassfish Server

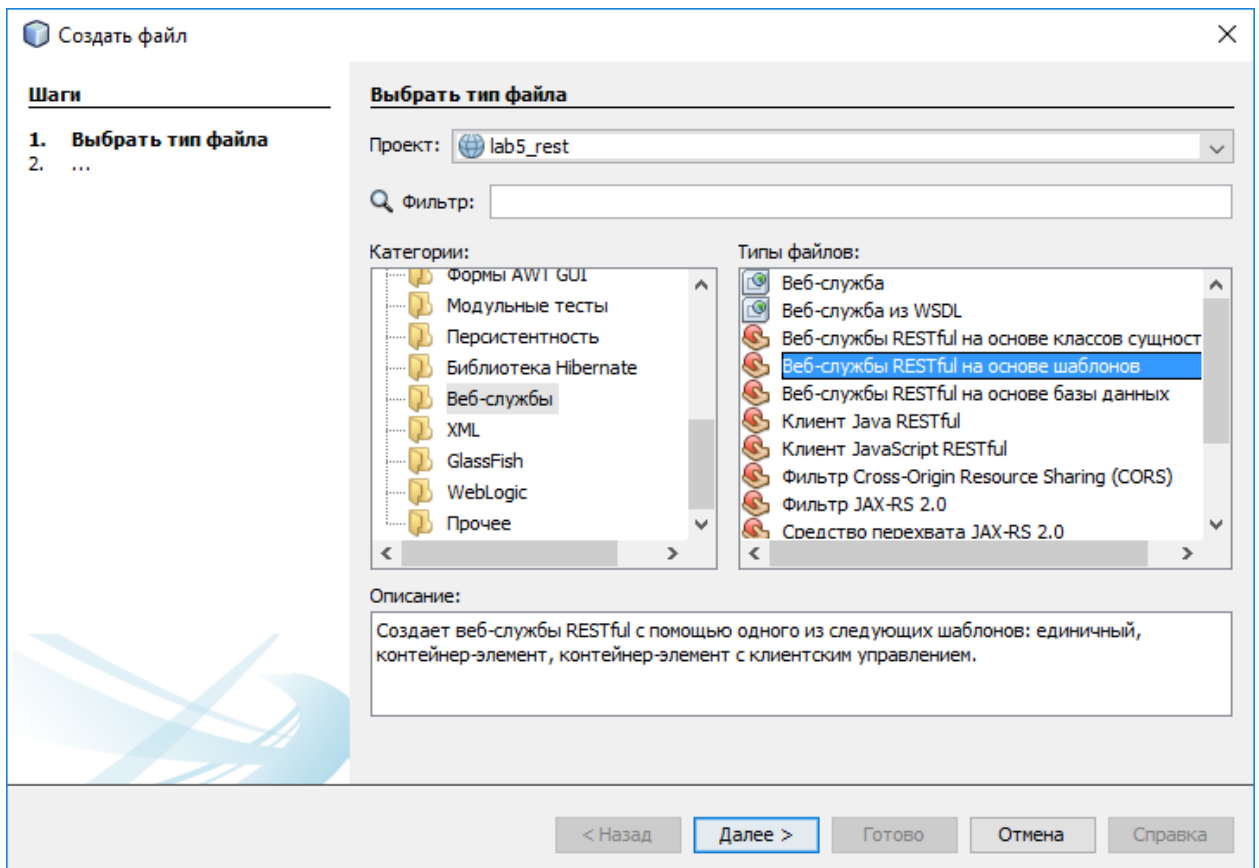
Ход выполнения работы

Рассмотрим вариант разработки веб-сервиса в виде Spring MVC контроллера.

1. Создать пустое веб приложение (Файл -> Создать проект -> Java Web -> Веб приложение)

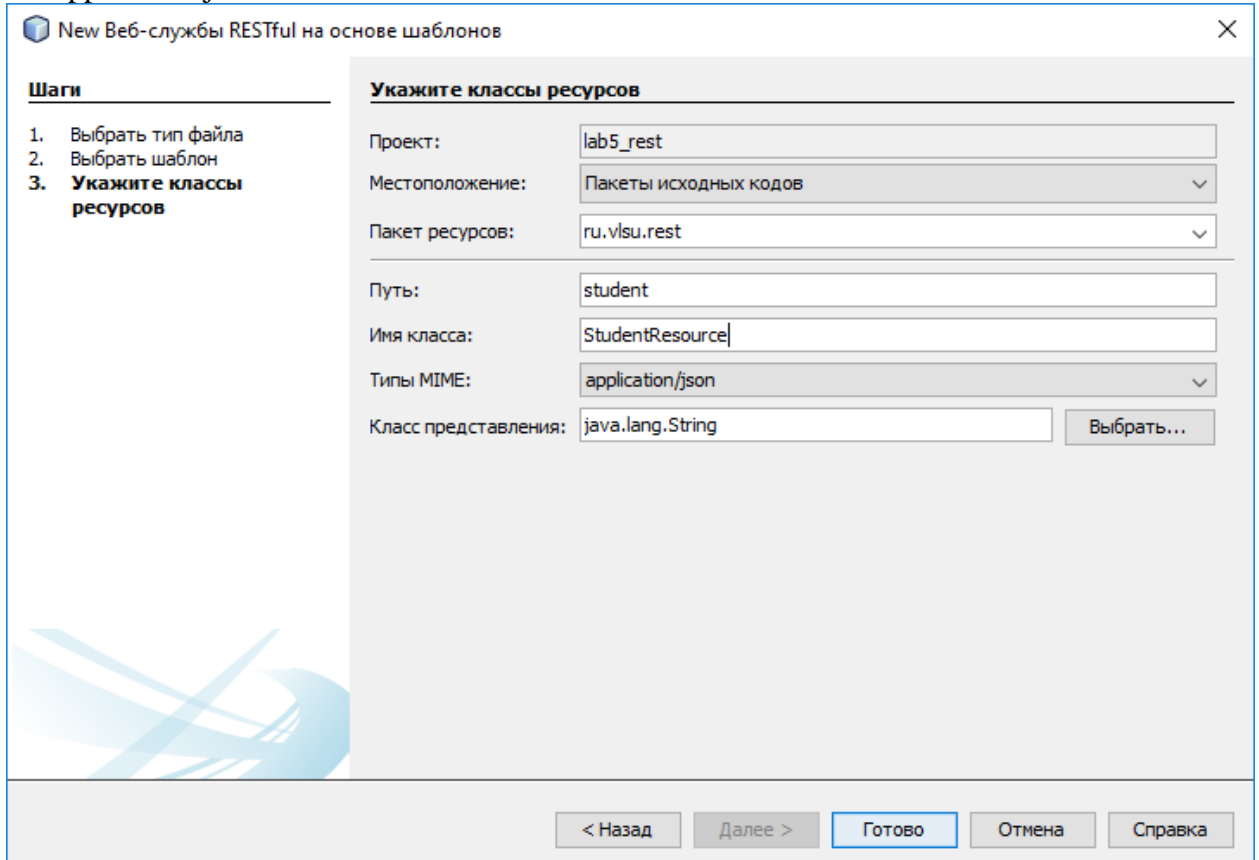


2. На созданном приложении нажать правой кнопкой мыши, выбрать Новый->Другое. В появившемся меню найти Веб-службы – Веб-службы Rest на основе шаблонов.



3. Нажать Далее, выбрать «Простой корневой ресурс». Нажать Далее.

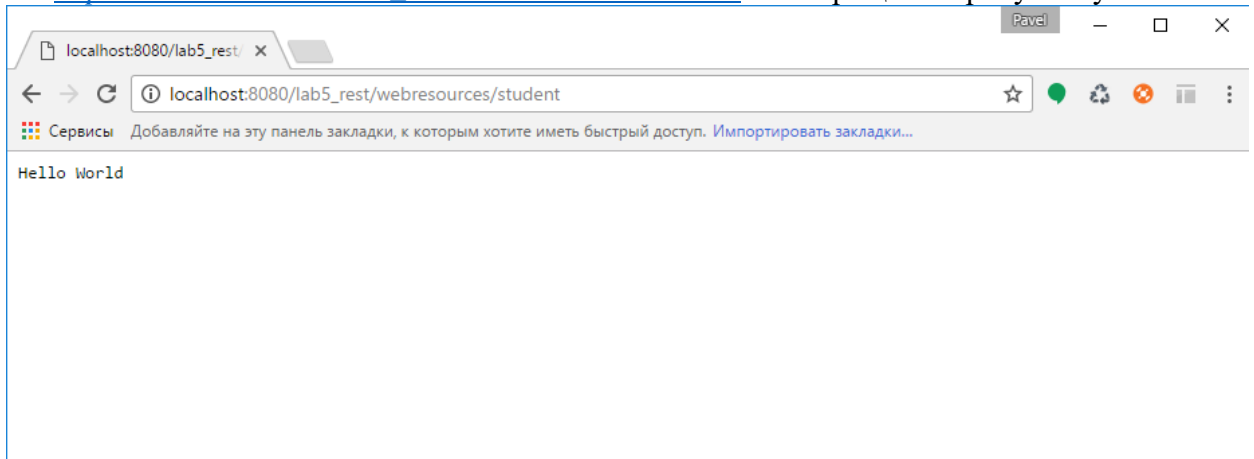
4. В появившемся окне указать Пакет ресурсов, путь, имя класса, Тип MIME указать application/json. Нажать Готово.



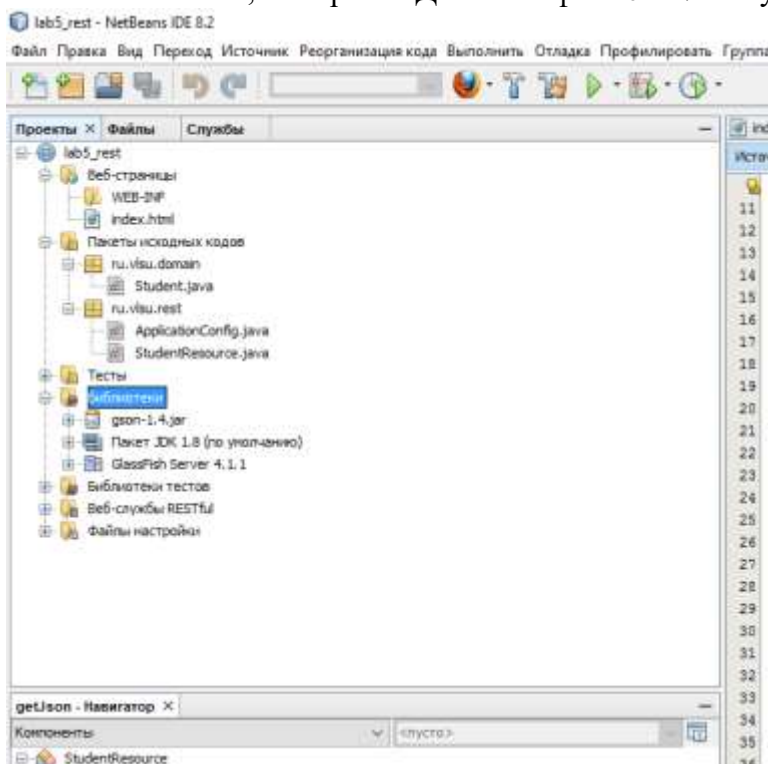
5. Изучить сгенерированные классы: StudentResource, ApplicationConfig.

6. Изменить метод getJson класса StudentResource так, чтобы он возвращал строку.

7. Запустить приложение. Сервис должен быть доступен по адресу http://localhost:8080/lab5_rest/webresources/student и возвращать строку из пункта 6.



8. Создайте класс Student, содержащий минимум три поля типа Integer, String, Date.
9. Добавьте в проект библиотек gson-1.4.jar. Для этого нажмите правой кнопкой на «Библиотеки», выберите «Добавить файл JAR/папку». Выберите файл gson-1.4.jar.



10. Сделайте так, что б метод getJson возвращал объект класса Student в формате JSON.

@GET

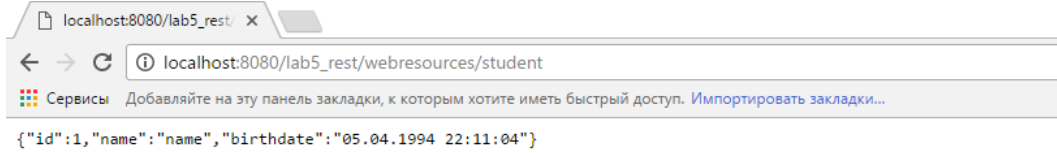
@Produces(MediaType.APPLICATION_JSON)

```
public String getJson() {  
    Gson gson = new Gson();  
    Student student = new Student();  
    student.setId(1);  
    student.setName("name");  
    Calendar c = Calendar.getInstance();  
    c.set(Calendar.YEAR, 1994);  
    c.set(Calendar.MONTH, 3);  
    c.set(Calendar.DAY_OF_MONTH, 5);  
}
```



```
student.setBirthdate(c.getTime());  
return gson.toJson(student);  
}
```

11. Проверьте результат работы приложения.



12. Создайте еще один REST сервис, который по GET запросу будет возвращать коллекцию (`java.util.List`) объектов типа `Student` (или другого типа) в формате JSON.

Контрольные вопросы

1. Поясните сделанный Вами выбор варианта реализации REST сервиса.
2. Приведите соответствие методов HTTP-запросов и действий с ресурсами, выполняемыми веб-сервисом.
3. Поясните структуру ответов сервиса.
4. Какие методы кроме GET поддерживаются протоколом HTTP?

Работа №6. Разработка веб-сервиса средствами JAX-WS

Задание

1. Разработать сессионный компонент без состояния с представлением в виде веб-сервиса. Использовать средства JAX-WS.

Средства выполнения

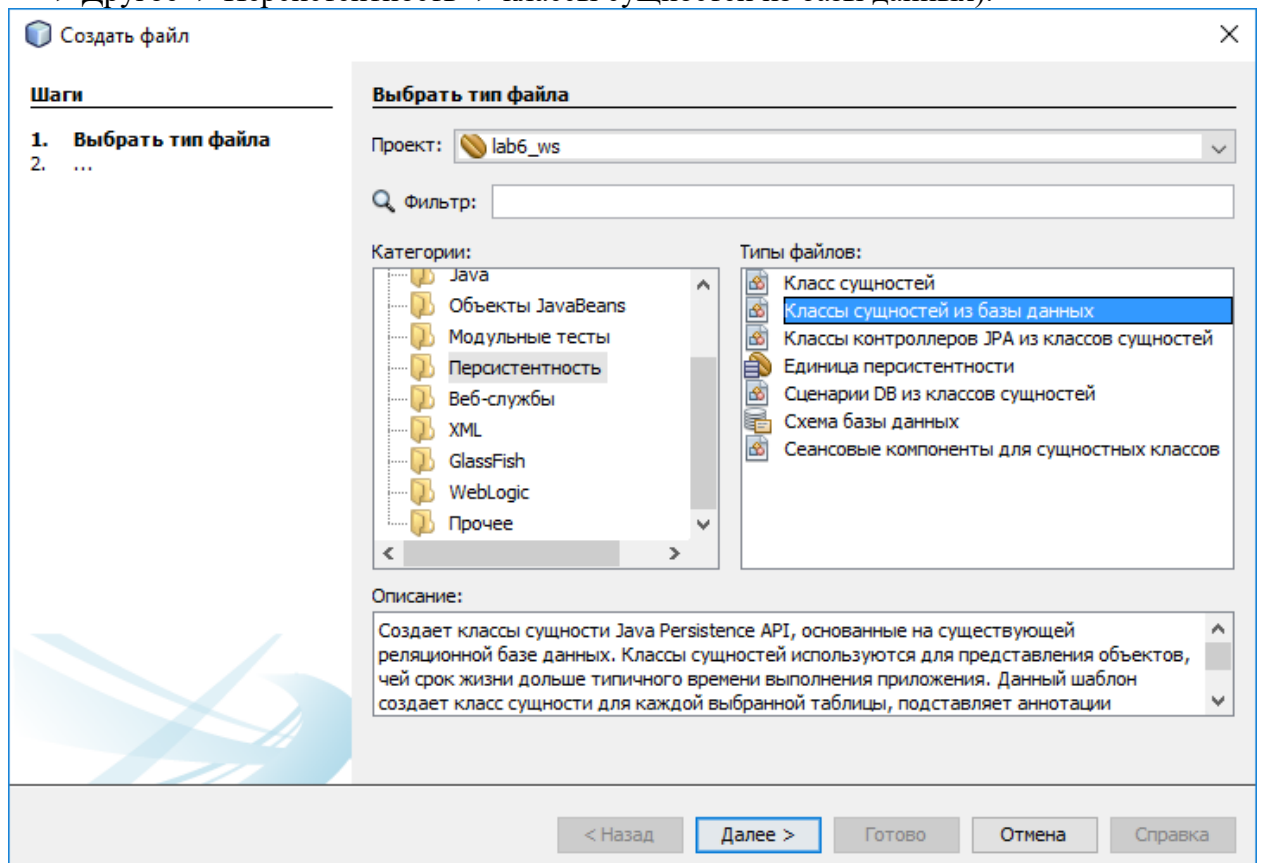
3. JDK 1.7
4. NetBeans IDE 8.2
5. Glassfish Server 4.0

Вопросы для допуска

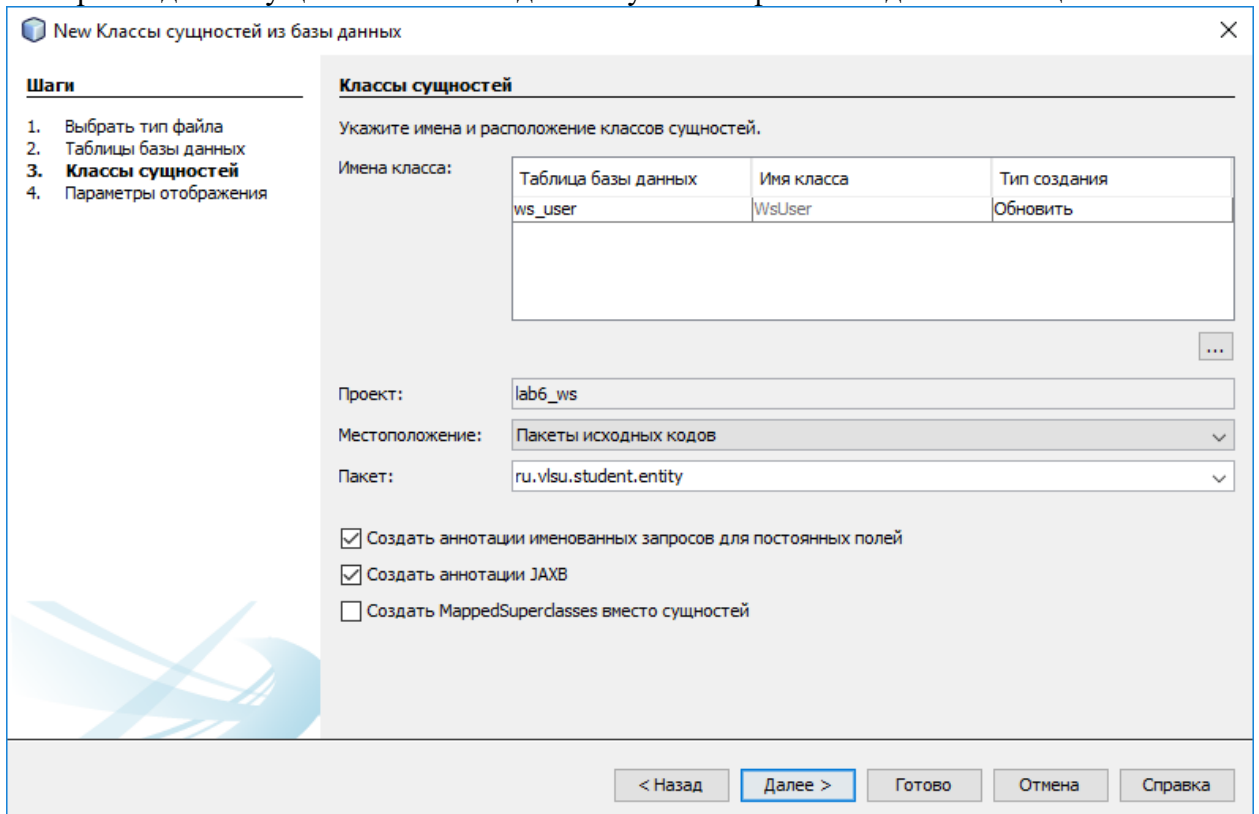
1. Какая XML-схема будет использоваться для описания структуры сообщений?
2. Какие операции будет реализовывать веб-сервис?
3. Какие сообщения об ошибках будет возвращать веб-сервис?

Ход выполнения работы

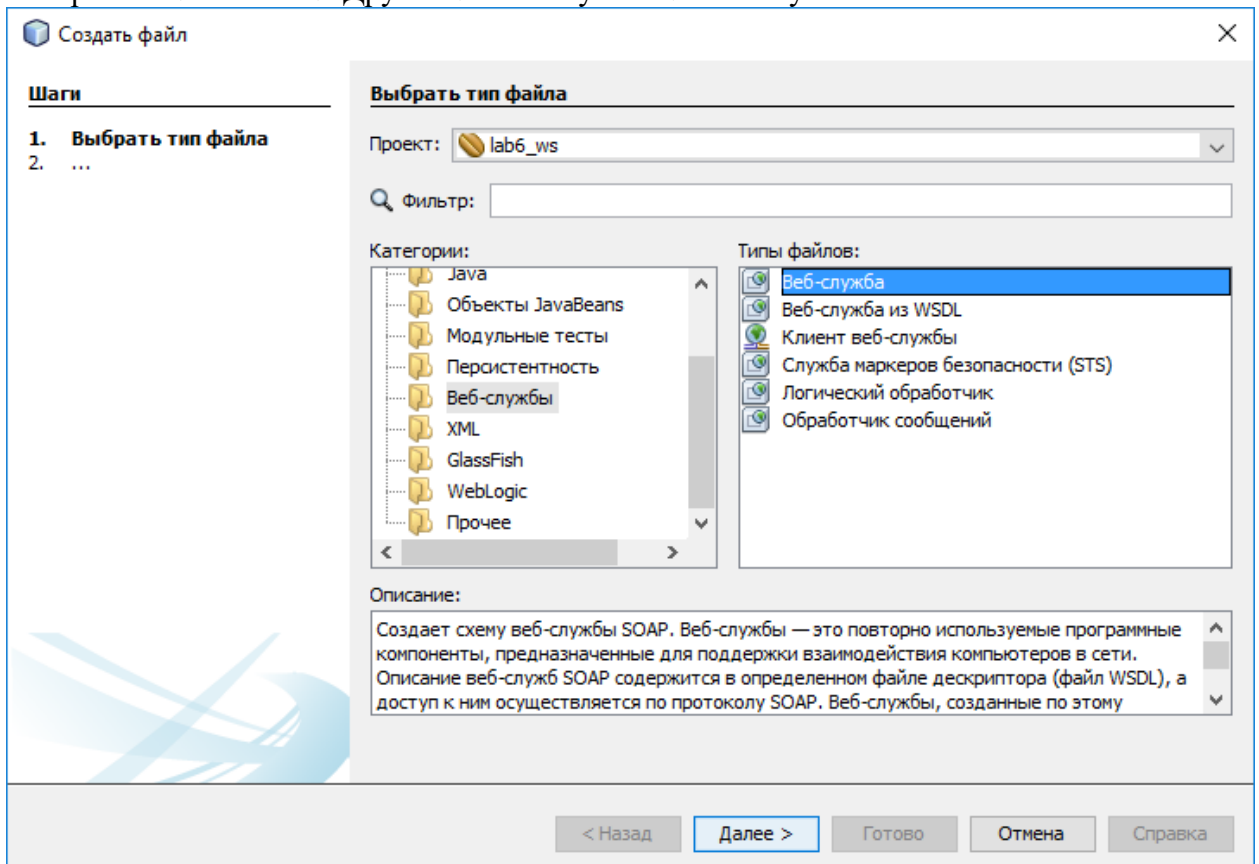
1. Создайте проект EJB-модуля. В качестве реализации веб-сервиса будет выступать EJB-компонент. Для создания проекта выберите пункт меню Новый проект, в списке Категории выберите категорию Java EE, а в списке Проекты вид проекта Модуль EJB. Нажмите Далее. Установите название проекта и его. Убедитесь, что выбрана версия платформы Java EE 7.
2. Сгенерируйте классы сущности из базы данных (контекстное меню проекта -> Новый -> Другое -> Персистентность -> классы сущностей из базы данных).



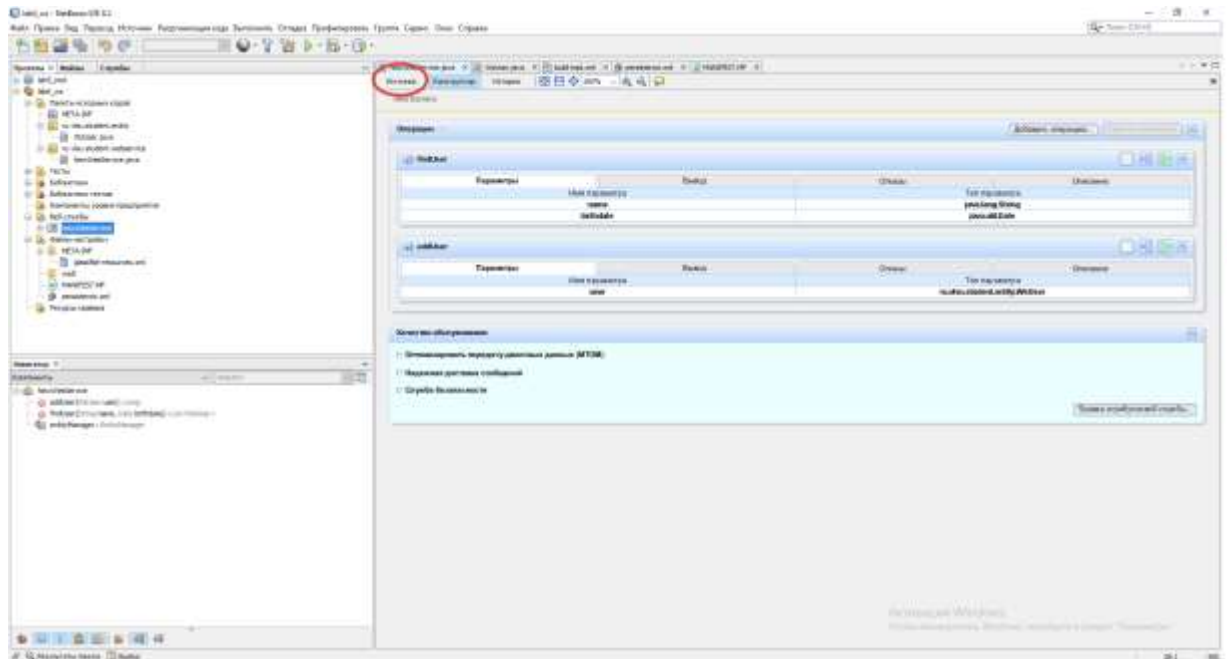
3. При создании сущностей из базы данных укажите флаг «Создать аннотации JAXB»



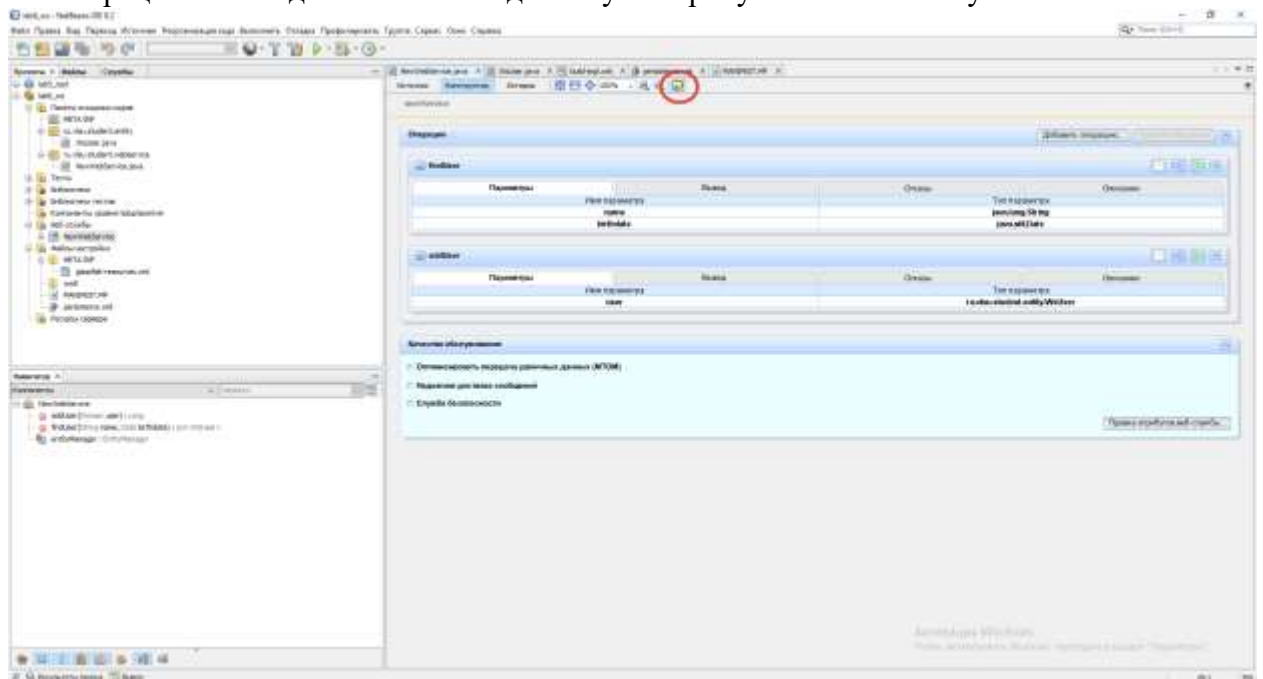
4. Создайте EJB-компонент веб сервиса. Для этого перейдите в контекстное меню проекта -> Новый -> Другое -> Веб-службы -> Веб служба.



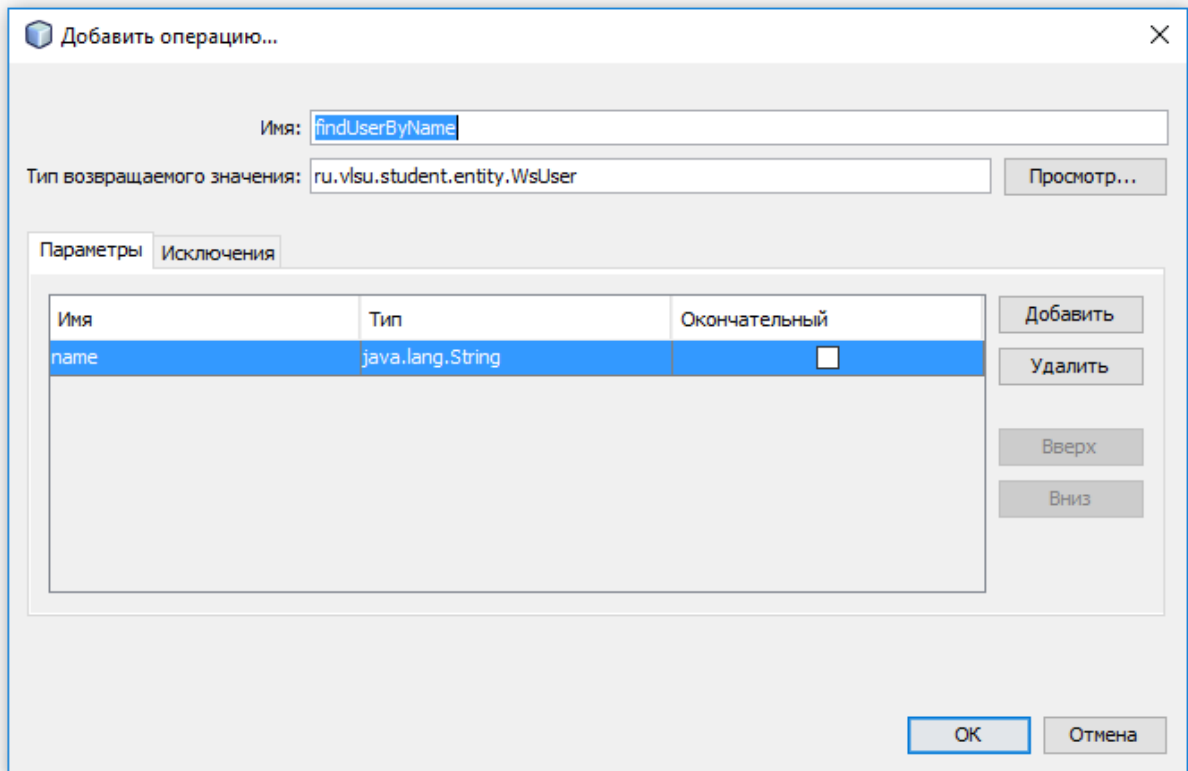
5. В появившемся окне появится мастер создания веб-службы. Для перехода к исходному коду необходимо нажать «Источник»



6. При помощи мастера или в исходном коде необходимо добавить 2 метода. Один для поиска коллекции сущностей; второй для сохранения сущности. Для добавления операции необходимо нажать выделенную на рисунке ниже кнопку.



7. В появившемся окне необходимо указать название операции, полноквалифицированный тип возвращаемого и тип принимаемого параметров.



8. Для реализации логики работы веб-службы необходимо перейти к исходному коду веб-службы, нажав «Источник».
9. Используя EntityManager для работы с базой данных выполнить реализацию необходимых операций.
10. Запустить приложение на Glassfish.
11. Через консоль администрирования Glassfish проверить, что приложение установлено.



12. Проверить работу сервиса



Перейти по ссылке <service url>/?Tester

13. Проверить работу свервиса используя SoapUI.

Контрольные вопросы

1. В чем состоят достоинства подхода к разработке веб-сервисов "от WSDL"?
2. Из каких частей состоит WSDL-описание веб-сервиса?
3. Приведите примеры запросов и ответов для реализованных операций веб-сервиса.
4. Какие аннотации JAX-WS используются в сессионном компоненте без состояния и для чего?
5. Какие аннотации JAX-WS используются в клиентском приложении и для чего?
6. Какой способ взаимодействия с сервисом используется в клиентском приложении и каковы его основные особенности?