

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет «Запорізька політехніка»

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт з дисципліни

«Архітектура і технології Web-служб»

для студентів спеціальності 123 «Комп'ютерна інженерія»

всіх форм навчання

Частина 2

XML-RPC Web-служби

2019

Методичні вказівки до лабораторних робіт з дисципліни «Архітектури і технології Web-служб» для студентів спеціальності 123 «Комп'ютерна інженерія» всіх форм навчання. Частина 2. XML-RPC Web-служби / Укл.: С.Д. Точилін. – Запоріжжя: НУ «Запорізька політехніка», 2019. – 14 с.

Укладачі:

С.Д. Точилін, доцент, к.ф-м.н.

Рецензент:

І.Я. Зеленьова, доцент, к.т.н.

Відповідальний за випуск:

С.Д. Точилін, доцент, к.ф-м.н.

Затверджено
на засіданні кафедри
«Комп'ютерних систем та мереж»
Протокол № 1 від 21.08.2019

Рекомендовано до видання:
НМК факультету КНТ
Протокол № 1 від 23.08.2019

ЗМІСТ

1 ОПИС ПРОТОКОЛУ XML-RPC.....	4
2 АРАСНЕ XML-RPC. РОЗПОДІЛЕНИЙ ДОДАТОК 3 XML-RPC WEB-SERVICESOM	8
2.1 Клієнт Web-сервіса.....	8
2.2 Web-сервіс.....	9
2.3 Запуск додатка	10
3 ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ.....	11
4 КОНТРОЛЬНІ ПИТАННЯ	14
5 РЕКОМЕНДОВАНА ЛІТЕРАТУРА	14

Лабораторна робота №2 «XML-RPC Web-служба»

Мета роботи: познайомитися з специфікацією XML-RPC. Навчитися створювати розподілені додатки за допомогою Java реалізації XML-RPC.

1 ОПИС ПРОТОКОЛУ XML-RPC

У наш час технологія WebServices для створення XML-документів обміну інформацією широко використовує стандарти SOAP та XML-RPC.

XML-RPC був створений в 1999 р. та вважається попередником SOAP.

Специфікація протоколу XML-RPC проста, вона містить не більше двох десятків елементів XML.

Типи даних XML-RPC наведені в таблиці 1.1.

Таблиця 1.1 – Типи даних XML-RPC

Елемент	Тип
<i4> або <int>	Ціле 4-байтове число
<boolean>	0 (false) або 1 (true)
<string>	Рядок символів ASCII
<double>	8-байтове речовинне число вещественное
<date.Time.iso8601>	Дата та час у форматі CCYYMMDDTHH:MM:SS
<base64>	Рядок коду Base 64
<array>	Масив
<struct>	Структура

Базові типи завжди включені в елемент <value>. Масив визначається з використанням елемента <array>, який містить елемент <data>. Елемент <data> містить список змінних. Подібно іншим типам даних XML-RPC елемент <array> повинен бути укладений в елемент <value>.

Наприклад, масив, який містить 3 рядка:

```

<value>
  <array>
    <data>
      <value><string>This</string></value>
      <value><string>is</string></value>
      <value><string> XML-RPC</string></value>
    </data>
  </array>
</value>

```

Масив може також містити суміш різних типів:

```

<value>
  <array>
    <data>
      <value><boolean>1</boolean></value>
      <value><string>Apple</string></value>
      <value><double>9.9</double></value>
    </data>
  </array>
</value>

```

Можна створювати також і багатомірні масиви шляхом додавання масиву усередину іншого масиву.

Структура XML-RPC подібна до структури мови C. Вона складається з декількох членів, які описуються елементами <member>. У кожного члена є ім'я <name> - рядок символів та значення <value> будь-якого типу, яким може бути й масив або інша структура. Члени структури розрізняються по іменах, тому порядок їх запису не має значення.

Наприклад:

```

<value>
  <struct>
    < member>
      <name>Name</name>
      <value><string>Tom</string></value>
    </ member>
  </struct>
</value>

```

```

< member>
  <name>Age</name>
  <value><int>27</int></value>
</ member>
</ struct>
</value>

```

XML-RPC-запит є комбінацією XML-документа та заголовків HTTP. XML-документ визначає зміст XML-RPC-запиту та описує процедуру, що викликається, та її параметри.

Заголовки HTTP забезпечують передачу даних запиту по мережі. Кожний XML-RPC-запит містить тільки один XML-документ. Його кореневим елементом є елемент `<methodCall>`. Елемент `<methodCall>` у свою чергу містить елементи `<methodName>` та `<params>`. Елемент `<methodName>` ідентифікує ім'я викликуваної процедури. Елемент `<params>` містить список параметрів процедури та їх величин (елемент `<param>` і `<value>`).

Заголовки HTTP відображають параметри відправника та умісту. Наприклад XML-RPC-запит до вилученого методу `circleArea`, що має речовинний параметр (радіус) виглядає в такий спосіб:

```

POST /xmlrpc HTTP 1.0
User-Agent: myXMLRPCClient/1.0
Host: 192.168.124.2
Content-Type: text/xml
Content-Length: 169
<?xml version="1.0"?>
<methodCall>
  <methodName>circleArea</methodName>
  <params>
    <param>
      <value><double>2.41</double></value>
    </param>
  </params>
</methodCall>

```

Формат XML-RPC-відповіді подібний до формату XML-RPC-запиту з невеликими змінами. Так в XML-документі відсутні елементи

<methodCall> та <methodName>. При цьому, кореневий елемент <methodCall> замінюється елементом <methodResponse>.

Крім того, XML-RPC-відповідь:

- може містити лише один параметр;
- цей параметр може бути масивом або структурою, тому можлива передача багатьох величин;
- наявність параметра, що повертається, для XML-RPC-відповіді є необхідною умовою.

У той час, заголовки HTTP відповіді вказують, що для його формування використовується код 200 OK.

Крім того, заголовки XML-RPC-відповіді:

- містять вказівку на те, що підтримується як протокол HTTP 1.0, так і HTTP 1.1;
- містять заголовок Content-Type, який повинен мати значення - text/xml;
- містять заголовок Content-Length, який вказує довжину відповіді в байтах.

XML-RPC-відповідь для XML-RPC-запиту, наведеного в 1.2 має вигляд:

```
HTTP/1.1 200 OK
Date: Sat, 06 Oct 2001 23:20:04 GMT
Server: Apache.1.3.12 (Unix)
Connection: close
Content-Type: text/xml
Content-Length: 124

<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value><double>18.24668429131</double></value>
    </param>
  </params>
</methodResponse>
```

XML-RPC-виключення генеруються у вигляді XML-документа XML-RPC-відповіді. Воно виникає при наявності помилок під час

виконання процедур. У цьому випадку елемент `<methodResponse>` містить елемент `<fault>` замість елемента `</params>`.

При цьому, елемент `<fault>` має тільки один елемент `<value>`, що вказує на відсутність методу, який викликається.

Наприклад, XML-документ XML-RPC-відповіді з виключенням, має вигляд:

```
<?xml version="1.0"?>
<methodResponse>
  <fault>
    <value><string>No such method!</string></value>
  </fault>
</methodResponse>
```

2 АРАСНЕ XML-RPC. РОЗПОДІЛЕНИЙ ДОДАТОК З XML-RPC WEB-SERVICES

У наш час існує велика кількість програмних продуктів, що реалізують XML-RPC-протокол при розробці Web-сервісів.

У даній лабораторній роботі ми будемо використовувати програмний продукт Apache XML-RPC v.1.1 (<http://xml.apache.org/xmlrpc>), який дозволяє створювати XML-RPC Web-сервіси мовою програмування Java.

Створимо Web-сервіс JavaWebService, що надає одну Web-послугу `sum()`, що повертає клієнтові суму двох чисел переданих при XML-RPC-запиті клієнта.

2.1 Клієнт Web-сервіса

Далі наведений код клієнта Web-сервіса:

```
import java.util.*;
import org.apache.xmlrpc.*;

public class JavaClient
{
  public static void main (String [] args)
  {
    try {
```

```

XmlRpcClient server = new XmlRpcClient("http://localhost:8080");
Vector params = new Vector();
int a=0; int b=0;
try{
    a = Integer.parseInt(args[0]);
    b = Integer.parseInt(args[1]);
}
catch(Exception e){System.out.println("Error: "+e);}
params.addElement(a);
params.addElement(b);
Object result = server.execute("sample.sum", params);
int sum = ((Integer) result).intValue();
System.out.println("The sum is: " + sum);
} catch (Exception exception) { System.err.println("JavaClient: " +
exception); }
}
}

```

Розглянемо більш докладно код XML-RPC-клієнта. Java пакет org.apache.xmlrpc містить класи необхідні для створення XML-RPC-клієнта та сервера. Пакет java.util необхідний для створення екземпляра класу Vector. Метод server.execute() надсилає запит серверу. У лістингу "sample" указує на заголовок, що визначає Web-сервіс.

Адреса Web-сервіса утримується в конструкторі класу XmlRpcClient.

2.2 Web-сервіс

Клас JavaWebService Web-сервіса, що містить метод sum(), наведений нижче.

```

import org.apache.xmlrpc.*;

public class JavaWebService
{
    public int sum(int x, int y)
    {
        return x+y;
    }
}

```

```

public static void main (String [] args)
{
    try {
        System.out.println("Attempting to start XML-RPC Web Service...");
        WebServer server = new WebServer(8080);
        server.addHandler("sample", new JavaWebService());
        server.start();
        System.out.println("Started successfully.");
        System.out.println("Accepting requests. (Halt program to stop.)");
    }
    catch (Exception exception) { System.err.println("Java Web Service: " +
exception); }
}
}

```

Відзначимо, що заголовок "sample" асоціюється з Java-класом Web-сервіса. Значення порту сервера утримується в конструкторі класу WebServer.

2.3 Запуск додатка

1. Створіть окрему директорію для розподіленого додатка. Помістіть в нього файли JavaClient.java і JavaWebService.java.
2. Відкомпілюйте файли за допомогою команди:

```
> javac *.java
```

3. Запустить Web-сервіс, виконавши команду:

```
> java JavaWebService
```

4. В окремій консолі запускать клієнта Web-сервіса, виконавши команду:

```
> java JavaClient 4 5
```

У результаті на консоль буде виведене повідомлення:

The sum is: 9

3 ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ

Завдання 1 – Виконайте додаток описаний в п. 2.

Завдання 2 – Виконайте додаток описаний в п. 2, з використанням утиліти **tcrtop**.

Завдання 3 – Створіть розподілений Java-додаток XML-RPC Web-сервіс якого при запиті клієнта робить обчислення відповідно до завдання Вашого варіанта. Клієнт передає Web-сервісу чисельні значення необхідних змінних.

Варіант 1. Знайти число членів нескінченного ряду:

$$\frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}, \dots, \frac{k+1}{k+2}, \dots, \text{менших деякого числа } \varepsilon.$$

Варіант 2. По заданій формулі $\frac{1}{(2k-1)(3k+1)}$ члена ряду з номером k скласти програму обчислення суми всіх членів ряду, не менших заданого числа ε .

Варіант 3. Дано дійсні числа x, ε ($0 < x \leq 1, \varepsilon > 0$). Обчислити з точністю ε :

$$\sum_{k=0}^{\infty} \frac{(-1)^k x^{2k+3}}{k!(2k+3)}.$$

Уважати, що необхідна точність досягнута, якщо черговий доданок виявився по модулю менше, ніж ε , - цей та всі наступні доданки можна вже не враховувати.

Варіант 4. По заданій формулі $\frac{(k+0,5)^2+1}{k}$ члена ряду з номером k скласти програму обчислення суми всіх членів ряду, не більших заданого числа ε .

Варіант 5. Дано дійсні числа x, ε ($0 < x \leq 1, \varepsilon > 0$). Обчислити з точністю ε :

$$\sum_{k=0}^{\infty} \frac{(-1)^k x^{3k+1}}{(2k)!(3k+1)}.$$

Вважати, що необхідна точність досягнута, якщо черговий доданок виявився по модулю менше, ніж ε , - цей та всі наступні доданки можна вже не враховувати.

Варіант 6. Дано дійсні числа x, ε ($x \neq 0, \varepsilon > 0$). Обчислити з

точністю ε :
$$\sum_{k=0}^{\infty} \frac{\sqrt[k]{|x|}}{k}.$$

Уважати, що необхідна точність досягнута, якщо черговий доданок, виявився, по модулю менше, ніж ε , - цей та всі наступні доданки можна не враховувати.

Варіант 7. Дано дійсні числа x, ε ($x \neq 0, \varepsilon > 0$). Послідовність $a_1, a_2, \dots$ утворена за наступним законом: $a_n = \frac{x^n}{(2n+1)!}$. Одержати

$a_1 + \dots + a_k$, де k - найменше ціле число, що задовольняє двом умовам: $k > 0$ і $|a_{k+1}| < \varepsilon$. Уважати, що необхідна точність досягнута, якщо черговий доданок виявився по модулю менше, ніж ε , - цей та всі наступні доданки можна вже не враховувати.

Варіант 8. Дано дійсні числа x, ε ($0 < x \leq 1, \varepsilon > 0$). Обчислити з точністю ε нескінченну суму та указати кількість врахованих

доданків: $\sum_{k=0}^{\infty} \frac{(-x)^{2k}}{2,5k!}$. Уважати, що необхідна точність досягнута,

якщо черговий доданок виявився по модулю менше, ніж ε , - цей та всі наступні доданки можна вже не враховувати.

Варіант 9. Дано дійсне число $\varepsilon > 0$. Послідовність $a_1, a_2, \dots$ утворена

за наступним законом: $a = \frac{n}{\sqrt{n^2+2}-\sqrt{n^2-1}}$. Знайти перший член

a_n ($n \geq 2$), для якого виконана умова $|a_n - a_{n-1}| > \varepsilon$.

Варіант 10. По заданій формулі $\frac{(2k^2+1)k}{2k+3}$ члена ряду з номером k

скласти програму обчислення суми всіх членів ряду, не більших заданого числа ε .

Варіант 11. Обчислити наближене значення нескінченної суми (праворуч від суми дається її точне значення, з яким можна зрівняти отриманий результат):

$$1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \dots \qquad \frac{\pi^2}{6}$$

Потрібне наближення вважається отриманим, якщо обчислено суму декількох перших доданків та черговий доданок виявився по модулі менше даного позитивного числа ε .

Варіант 12. Дано дійсні числа a, b, ε ($a > b > 0, \varepsilon > 0$).

Послідовності $x_1, x_2, \dots, y_1, y_2$ утворені за законом:

$$x_1 = a, y_1 = b, x_k = \frac{1}{2,5}(x_{k-1} + y_{k-1}), y_k = \sqrt{x_{k-1}y_{k-1}}. \text{ Знайти}$$

перше x_n , таке, що $|x_n - y_n| < \varepsilon$.

Варіант 13. Дано дійсні числа x, ε ($x \geq 1, \varepsilon > 0$). Обчислити з

точністю ε :
$$\sum_{k=1}^{\infty} \frac{1}{xk^2}.$$

Вважати, що необхідна точність досягнута, якщо черговий доданок виявився по модулю менше, ніж ε , - цей та всі наступні доданки можна вже не враховувати.

Варіант 14. Дано дійсні числа x, ε ($x \neq 0, \varepsilon > 0$). Обчислити з точністю ε :

$$\sum_{k=0}^{\infty} \frac{x^{2k}}{2k!}$$

Вважати, що необхідна точність досягнута, якщо черговий доданок виявився по модулю менше, ніж ε , - цей та всі наступні доданки можна вже не враховувати.

Варіант 15. Обчислити наближене значення нескінченної суми (праворуч від суми дається її точне значення, з яким можна зрівняти отриману відповідь):

$$\frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + \frac{1}{3 \cdot 4} + \dots + K \qquad 1$$

Потрібне наближення вважається отриманим, якщо обчислено суму декількох перших, що складаються, і черговий доданок виявився по модулі менше даного позитивного числа ε .

4 КОНТРОЛЬНІ ПИТАННЯ

1. Які типи даних може використовувати XML-RPC-протокол?
2. Який формат має XML-RPC-запит?
3. Який формат має XML-RPC-відповідь?
4. Яким чином генеруються XML-RPC-виключення?
5. Що являє собою Apache XML-RPC?

5 РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Хабибулин И.Ш. Создание распределенных приложений на Java 2. – СПб.: БХВ-Петербург, 2002. – 704 с.
2. Хабибулин И.Ш. Самоучитель Java. – СПб.: БХВ-Петербург, 2001. – 464 с.
3. Хабибулин И.Ш. Создание Web-служб средствами Java. – СПб.: БХВ-Петербург, 2003. – 400 с.
4. Laurent S. St., Johnston J., Dumbill E. Programming Web Services with XML-RPC: Creating Web Application Gateways. – O'Reilly, 2001. – 230 p.
5. Cerami E. Web Services Essentials: Distributed Applications with XML-RPC, SOAP, UDDI & WSDL. – O'Reilly, 2002. – 308 p.