

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 1

SOAP 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 1. SOAP Web-служби / Укл.: С.Д. Точилін. – Запоріжжя: НУ «Запорізька політехніка», 2019. – 18 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. ЗАГАЛЬНІ ВІДОМОСТІ	4
1.1 Web-сервіси.....	4
1.2 Протокол SOAP	4
1.3 AXIS.....	6
2 СТВОРЕННЯ РОЗПОДІЛЕНОГО ДОДАТКУ З WEB-SERVICES.....	6
2.1 Створення Web-сервіса для Axis.....	6
2.2 Клієнт Web-сервіса.....	7
2.3 Запуск додатка	8
2.4 TCP-монітор.....	8
3 ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ.....	13
4 КОНТРОЛЬНІ ПИТАННЯ	15
5 ЗМІСТ ПИСЬМОВОГО ЗВІТУ	15
6 РЕКОМЕНДОВАНА ЛІТЕРАТУРА	16
Додаток А	16
Додаток Б.....	17

Лабораторна робота №1 «SOAP Web-служба»

Мета роботи: навчитися створювати розподілені додатки за допомогою Java реалізації SOAP.

1. ЗАГАЛЬНІ ВІДОМОСТІ

1.1 Web-сервіси

Розподіленим називається додаток, окремі компоненти якого працюють на різних комп'ютерах і використовують різні мережеві засоби. Однак вони взаємодіють так, що додаток виглядає як єдине ціле, начебто всі його компоненти розташовані на одній машині.

В наш час широке розповсюдження отримали розподілені додатки на основі Web-сервісів (Web-служб).

Web-сервіс - це додаток, який функціонує за допомогою стандартних Інтернет-технологій та ідентифікується рядком URI (відмінність URL від URI у тому, що URL - це URI, який крім ідентифікації ресурсу вказує його місцезнаходження), доступ до якого здійснюється за допомогою стандартних протоколів (HTTP, SMTP, FTP, XML-RPC, SOAP та ін.).

Розподілений додаток - компоненти якого жорстко прив'язані до обраної технології та певної платформи називається *тісно зв'язаним*

Розподілений додаток, компоненти якого можуть працювати на різних платформах та замінювати один одного, називається *слабко зв'язаним* додатком.

Web-сервіси відносяться до слабо зв'язаних додатків.

1.2 Протокол SOAP

Протокол SOAP виник в 1998 році у співпраці фірми UserLand та корпорації Microsoft, але потім його розробка була передана в консорціум W3C, який готує зараз рекомендації з його застосування.

Протокол SOAP не розрізняє виклик процедури та відповідь на нього, а просто визначає формат послання (message) у вигляді документа XML. Послання може містити виклик процедури, відповідь на нього, запит на виконання якихось інших дій або просто текст.

Специфікацію SOAP не цікавить зміст послання, вона задає тільки його оформлення.

Кореневий елемент документа, що посилається, <Envelope> містить необов'язковий заголовок <Header> та обов'язкове тіло <Body>. Схема SOAP-послання така:

```
<?xml version='1.0' ?>
<env:Envelope
    xmlns:env="http://www.w3.org/2002/06/ soap-envelope">
    <env:Header>
        <!-- Блоки заголовка -->
    </env:Header>
    <env:Body>
        <!-- Зміст послання -->
    </env:Body>
</env:Envelope>
```

У заголовку міститься один або кілька блоків, оформлення та зміст яких ніяк не регламентуються. Точно так само нічого не говориться про зміст тіла послання. Проте, розрізняють процедурний стиль послання SOAP, призначений для виклику вилучених процедур, та документний стиль, призначений для обміну документами XML. Процедурний стиль часто називають RPC-стилем, а документний стиль - XML-стилем.

При процедурному стилі послання SOAP при виклику вилучених процедур у заголовку послання встановлюють параметри виклику, наприклад, номер транзакції, у якій виконується процедура. У тілі послання перераховуються аргументи викликуваної процедури.

При документному стилі послання SOAP послання містить не аргументи виклику якоїсь процедури, а просто документ XML або структуру даних.

1.3 AXIS

У наш час є багато засобів створення SOAP-серверів та SOAP-клієнтів. Їх довгий список можна подивитися на сайті <http://www.software.org/>.

Провідні фірми, що випускають продукти для надання Web-послуг - IBM, Microsoft, Oracle - випустили свої SOAP-продукти.

Фірма IBM спочатку випустила продукт розробки SOAP-серверів та клієнтів SOAP4J, але потім передала його розробку в співтовариство Apache Software Foundation, що випустило на його основі набір класів та інтерфейсів Apache SOAP (<http://xml.apache.org/SOAP/>). Цей продукт входив до складу сервера додатків IBM WebSphere і дотепер входить до складу багатьох серверів додатків.

Надалі співтовариство Apache SOAP стало випускати програмні продукти Axis та Axis2 (<http://xml.apache.org>), які також реалізують протокол SOAP. Axis2 має більше функціональних можливостей. Більш простий Axis характеризується високою стабільністю та використовується при побудові Web-сервісів і у наш час. Крім того він входить до складу сервера додатків IBM WebSphere Application Server та багатьох інших серверів додатків.

До складу Axis входить SOAP-сервер - Java-клас SimpleAxisServer.

2 СТВОРЕННЯ РОЗПОДІЛЕНОГО ДОДАТКУ З WEB-SERVICOM

2.1 Створення Web-сервіса для Axis

Для створення SOAP Web-сервіса, що буде працювати під керуванням Axis, досить написати класи Java, що реалізують Web-сервіс, дати файлу, що містить вихідні тексти цих класів, розширення ".jws", та покласти цей файл у каталог axis сервера додатків (у нашій випадку - **C:\axis\webapps\axis**).

Створимо Web-сервіс AddService, який надає одну Web-послугу add(), що повертає клієнтові суму двох чисел переданих при SOAP-запиті клієнта. Клас AddService, що містить метод add(), наведений нижче.

```
public class AddService
{
    public int add(int a, int b)
    {
        int sum=a+b;
        return sum;
    }
}
```

Назвемо даний файл `AddService.jws` та покладемо його в каталог `axis` сервера додатків (`C:\axis\webapps\axis`). Після цього Web-сервіс готов до використання. Не потрібно компілювати клас `AddService`, Axis зробить це сам при першому запиті до Web-сервіса.

2.2 Клієнт Web-сервіса

Далі наведений код клієнта Web-сервіса `AddService`.

```
import org.apache.axis.client.Call;
import org.apache.axis.client.Service;
import java.net.*;
public class AddServiceClient
{
    public static void main(String[] args) throws Exception
    {
        Service service = new Service();
        Call call = (Call)service.createCall();
        String endpoint = "http://localhost:8080/AddService.jws";
        call.setTargetEndpointAddress(new URL(endpoint));
        call.setOperationName("add");
        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);}
        int sum =(Integer)call.invoke(new Object[] {a,b});
        System.out.println("Sum "+a+ " + " + b+ " = "+ sum); }}
```

Використані інтерфейси `service` та `call` пакета JAX-RPC. Вони реалізовані класами, що входять в Axis. Основну роль в одержанні Web-послуги грає об'єкт типу `call`. Він містить адресу Web-сервіса `endpoint` та назву Web-послуги `add`. У нього заносяться аргументи Web-послуги у вигляді масиву об'єктів.

У нашій простому прикладі масив аргументів складається тільки із двох елементів - двох цілих чисел `a` та `b`. Нарешті, об'єкт `call` методом `invoke` повертає об'єкт, що містить результат виконання Web-послуги.

2.3 Запуск додатка

У папці з Web-сервіса `AddService.jws` запустить `SimpleAxisServer`, виконавши команду:

```
> java org.apache.axis.transport.http.SimpleAxisServer
```

В окремій консолі запустить клієнта Web-сервіса, виконавши команду:

```
> java AddServiceClient 3 5
```

У результаті на консоль буде виведене повідомлення:

```
Sum 3 + 5 = 8
```

2.4 TCP-монітор

Axis поставляється з корисною утилітою **tcpmon** (TCP - монітор). Ця утиліта на основі пакета `Swing`, може використовуватися для установки TCP - тунелю та перегляду XML та HTML-документів, якими обмінюються клієнт та сервер розподіленого додатка.

Використання TCP - монітора продемонструємо на прикладі роботи Web-сервіса `AddService`. При цьому, виконаємо ряд кроків.

1. Запуск утиліти (запуск зробити у директорії де перебуває клієнт Web-сервіса):

```
> java org.apache.axis.utils.tcpmon
```

Ця команда повинна відобразити вікно, показане нижче, у якому ми можемо ввести значення, показані на рисунку (за замовчуванням текстові поля будуть порожніми):

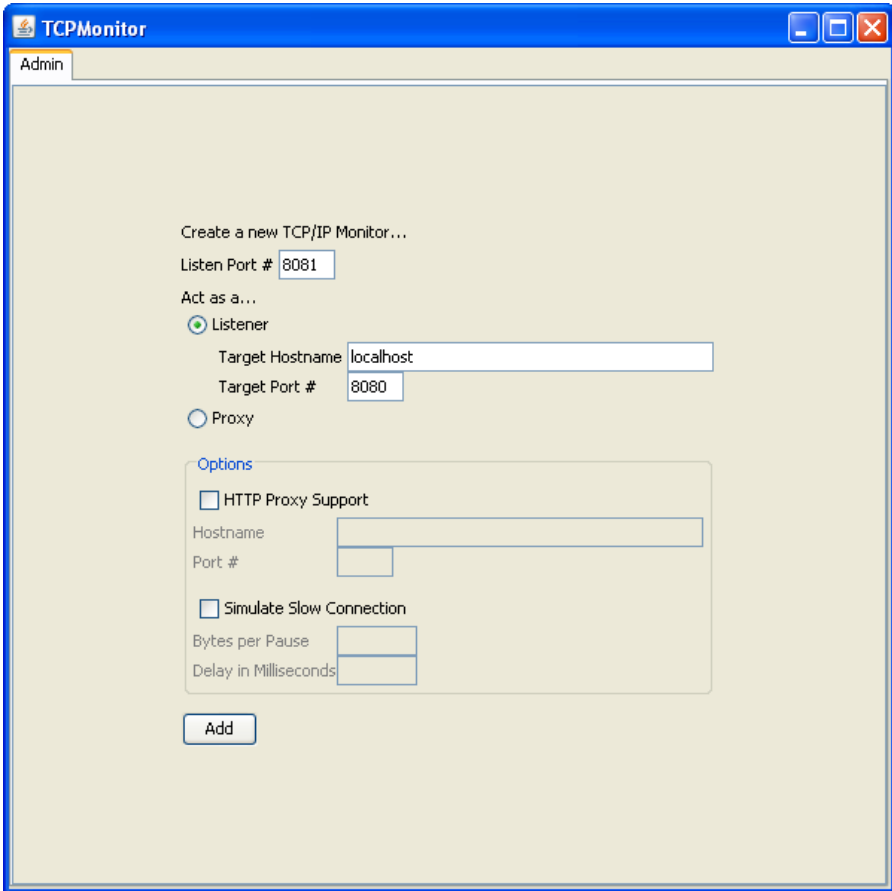


Рисунок 2.1 - TCP-монітор

Не використовуйте ті самі значення портів у полях *Listen Port* і *Target Port*, інакше ми ввійдемо в нескінченний цикл.

2. Після уведення значень, показаних на рисунку вище, натисніть кнопку *Add*. У результаті в основному вікні буде створена нова закладка з назвою *Port 8081*.

3. Натисніть на закладку *Port 8081*, щоб переглянути її вміст. У цьому випадку на екрані відобразиться наступне:

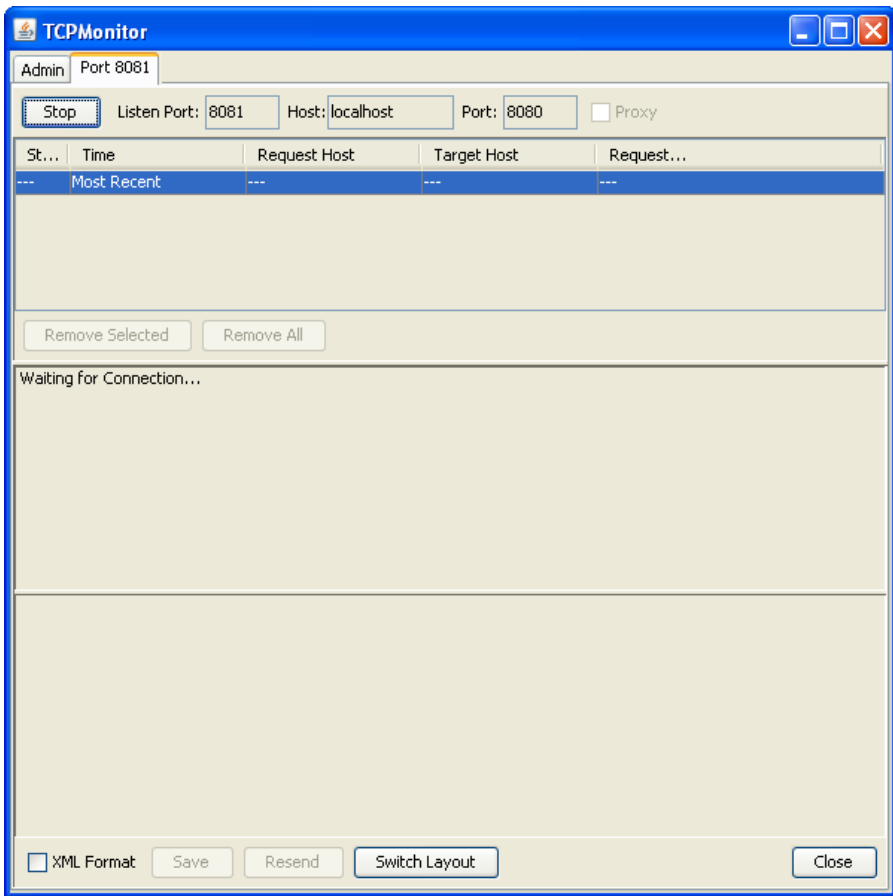


Рисунок 2.2 – Закладка *Port 8081*

4. Тому що ми встановили утиліту **tcpmon** на прослуховування порту 8081, нам потрібно змінити нашого клієнта так, щоб він передавав свої запити на порт 8081. Для цього нам потрібно змінити один рядок у коді нашого клієнта Web-сервіса AddService. У цих цілях

відкриємо файл *AddServiceClient.java* у текстовому редакторі й замінимо його рядок:

```
String endpoint = "http://localhost:8080/AddService.jws";  
на  
String endpoint = "http://localhost:8081/AddService.jws";
```

5. Файл JWS повинен залишитися незмінним. Запустить SimpleAxisServer. Скомпілюйте та запустите зміненого клієнта *AddServiceClient*. Після виконання клієнтського додатка вікно утиліти **tcpmon** повинне виглядати в такий спосіб (ми можемо перемикатися між горизонтальним та вертикальним видом, використовуючи кнопку *Switch Layout*):

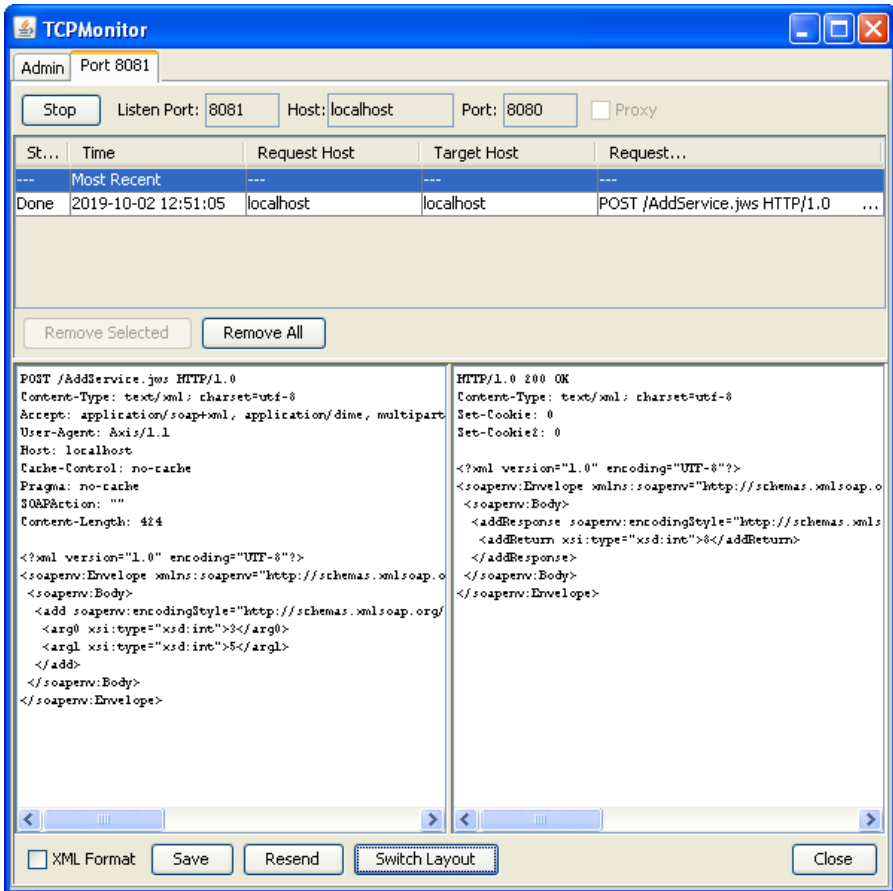


Рисунок 2.3 – TCP-монітор

Як видно з останнього рисунка, утиліта **tcpmon** дозволяє переглядати XML-документи, якими обмінюються Web-сервіс та його клієнт.

3 ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ

Завдання 1 - Виконаєте додаток описаний в п. 2.

Завдання 2 - Виконаєте додаток описаний в п. 2, з використанням утиліти **tcrcmon**.

Завдання 3 - Створіть розподілений Java-додаток Web-сервіс якого по запиті клієнта робить обчислення у відповідності математичною формулою Вашого варіанта. Клієнт передає серверу чисельні значення змінних зазначених у формулі. При виконанні роботи користуйтеся довідковими даними по математичним методам java, наведеним у додатку А та Б.

№ варіанта	Математична формула
1	$\frac{3 \cos^2 \left(x - \frac{\pi}{6} \right)}{\frac{1}{2} + \sin y^2}$
2	$\frac{ x - y }{(1 + 2x)^a} - e^{\sqrt{1+\omega}}$
3	$\frac{5a^{nx}}{b+c} - \sqrt{ \cos x^3 }$
4	$\ln a^7 + \arctg x^2 + \frac{\pi}{\sqrt{ a+x }}$
5	$\sqrt[5]{\frac{(a+b)^2}{c+d}} + e^{\sqrt{x+1}}$
6	$\frac{1}{2\pi} \sqrt{\frac{2mg}{m(a \sin \alpha + b \cos \alpha)}}$
7	$\frac{1}{4} \left(\frac{1+x^2}{1-x} + \frac{1}{2} \operatorname{tg} x \right)$

8	$a + \sqrt{b} + \ln x^3 + e^{\sqrt[3]{c}}$
9	$ \sin \alpha \cos \frac{\alpha}{2} + \sqrt{a^2 + b^2}$
10	$\sqrt[5]{x}(\operatorname{arctg} z + \cos^2 y)$
11	$\sqrt[4]{y} + \sqrt[3]{x - y}$
12	$e^{x \cos \frac{\pi}{4}} \cos(x \sin \frac{\pi}{4})$
13	$\sqrt{ \ln(1 - 2x \cos \frac{\pi}{3} + x^2) }$
14	$\sqrt{a_0 + a_1 x + a_2 \sqrt[3]{ \sin x }}$
15	$\frac{\operatorname{tg} x}{x} + \sqrt[3]{\sin x^3}$
16	$x^2 + y^2 + z/i + \sqrt{y}$
17	$10x + 5/y + 10/z + \sqrt[3]{m}$
18	$\operatorname{tg} x / (a + b) - \sqrt{ \sin y }$
19	$a^2 + bx^3 + xyz$
20	$\sqrt[3]{(a + b)^2 / c} + e^{0,75}$
21	$mgh + 0,5mv^2$
22	$\sqrt{ab + bc + c} + n^3$
23	$a + \sqrt[3]{b} + c^3 + \ln d $

4 КОНТРОЛЬНІ ПИТАННЯ

1. Який додаток називають розподіленим?
2. Що розуміють під тісно та слабо зв'язаним розподіленим додатком?
3. Дайте визначення Web-сервіса.
4. Що являє собою SOAP?
5. Який формат має SOAP-повідомлення?
6. Що являє собою Axis?
7. Що являє собою TCP - монітор?
8. XML. Яку структуру має XML документ?
9. Що розуміють під простором імен XML?
10. У яких цілях використовують схеми XML?

5 ЗМІСТ ПИСЬМОВОГО ЗВІТУ

Письмовий звіт повинен містити.

1. Тему та мету роботи.
2. Тексти файлів програми-клієнта та Web-сервіса, які були створені при рішенні завдання 3. Крім того, приведіть та поясніть XML-файли, якими обмінюються клієнт та сервер при виконанні завдання 3.
3. Відповіді на контрольні питання.
4. Висновки.

Письмовий звіт повинен бути оформлений на комп'ютері, мати титульну сторінку із зазначенням назви дисципліни, теми лабораторної роботи, автора звіту та викладача, дати складання письмового звіту. Далі на кожній сторінці звіту в колонтитулах має міститись прізвище, ім'я та номер групи студента (верхній колонтитул), тема лабораторної роботи та номер сторінки (нижній колонтитул). В іншому випадку звіт може бути оформлений в рукописному варіанті.

6 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. Машнин Т.С. Web-сервисы Java. – СПб.: БХВ-Петербург, 2012. – 560 с.
5. Hewitt E. Java SOA Cookbook. – Sebastopol, CA: O'Reilly Media, 2009. – 742 p.
6. Bean J. SOA and Web Services Interface Design: Principles, Techniques, and Standards. – Burlington, MA, USA: Elsevier Inc., 2010. – 360 p.

Додаток А

Java-методи для перетворення рядків (тип String) у числові типи даних

Метод	Тип аргументу x	Тип методу
Integer.parseInt(x)	String	int
Long.parseLong(x)	String	long
Float.parseFloat(x)	String	float
Double.parseDouble(x)	String	double

Додаток Б

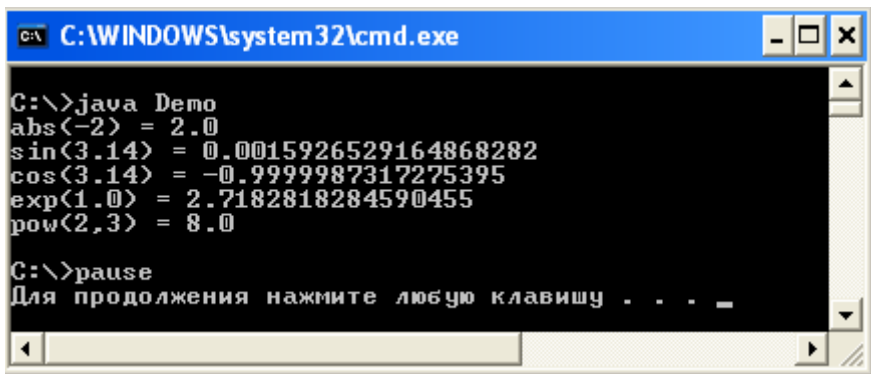
Убудовані математичні методи Java (клас Math)

Метод	Призначення	Тип аргументу	Тип методу
abs(x)	Обчислення абсолютного значення x	int long float double	int long float double
sin(x)	Обчислення синуса x	double	double
cos(x)	Обчислення косинуса x	double	double
tan(x)	Обчислення тангенса x	double	double
asin(x)	Обчислення арксинуса x	double	double
acos(x)	Обчислення арккосинуса x	double	double
atan(x)	Обчислення арктангенса x	double	double
sqrt(x)	Обчислення квадратного кореня з x	double	double
pow(x,y)	Зводить x у ступінь y	double	double
log(x)	Обчислення натурального логарифма x	double	double
exp(x)	Зводить e в заданий ступінь	double	double
min(x)	Повертає менше із двох чисел	int long float double	int long float double
max(x)	Повертає більше із двох чисел	int long float double	int long float double
round(x)	Округляє число до найближчого цілого	float double	int long
floor(x)	Округляє число до найближчого цілого з нестачею	double	double
ceil(x)	Округляє число до найближчого цілого з надлишком	double	double

Математичні методи Java містить клас `Math`. При цьому, їхній виклик має наступний синтаксис: *Math.метод_Java*

Приклад використання деяких математичних методів Java:

```
class Demo
{
    public static void main (String[] args)
    {
        double x, y, result;
        x=-2;
        result=Math.abs(x);
        System.out.println ("abs(-2) = " + result);
        x=3.14;
        result=Math.sin(x);
        System.out.println ("sin(3.14) = " + result);
        result=Math.cos(x);
        System.out.println ("cos(3.14) = " + result);
        x=1;
        result=Math.exp(x);
        System.out.println ("exp(1.0) = " + result);
        x=2; y=3;
        result=Math.pow(x,y);
        System.out.println ("pow(2,3) = " + result);
    }
}
```



The screenshot shows a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The command prompt displays the output of the Java program executed in the previous block. The output is as follows:

```
C:\>java Demo
abs(-2) = 2.0
sin(3.14) = 0.0015926529164868282
cos(3.14) = -0.9999987317275395
exp(1.0) = 2.7182818284590455
pow(2,3) = 8.0

C:\>pause
Для продолжения нажмите любую клавишу . . .
```