



Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційні систем та технологій

ЛАБОРАТОРНА РОБОТА №2

з дисципліни «Основи програмування - 2»

Тема: «Оброблення запитів у сервлетах»

Виконали:

студенти групи ІА-31
Клим'юк В.Л, Самелюк А.С,
Дук М.Д, Сакун Д.С

Перевірив:

асистент кафедри ІСТ
Степанов А. С.

Тема: Оброблення запитів у сервлетах

Мета: Мета лабораторної роботи полягає у вивченні та практичному опануванні основних методів оброблення запитів у сервлетах Java, зокрема отримання даних від клієнта, оброблення та аналіз отриманих даних, виконання дій відповідно до запиту та відправлення результатів клієнту.

Хід роботи

1. Пригадати як використовувати Servlet API для створення динамічних веб-сторінок.
2. Для свого варіанту з л/р №2 першого семестру створити сторінку з HTML-формою для введення значень параметрів обчислення формули. При натисканні кнопки «обчислити» параметри мають передаватись на сервлет, який відображає результати розрахунку або, у разі невірно введених параметрів чи виникнення помилки в процесі розрахунку, відображає повідомлення про помилку (Завдання №1, 2, 3).
3. За допомогою використання Cookie модифікувати завдання 2 таким чином, щоб при відкритті форми в полях відображались останні значення, які користувач вводив минулого разу (в тому числі після закриття вікна браузера), але не раніше ніж 2 дні тому.

```
package com.example.lab_02;

import java.io.*;

import jakarta.servlet.http.*;
import jakarta.servlet.annotation.*;

@WebServlet(name = "formulaServlet", value = "/formula-1")
public class FormulaServlet_1 extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException {response.setContentType("text/html");

        String aParam = request.getParameter("a");
        String bParam = request.getParameter("b");
        String cParam = request.getParameter("c");
        String dParam = request.getParameter("d");

        double a ;
```

```

double b ;
double c ;
double d;

try {
    a = Double.parseDouble(aParam);
    b = Double.parseDouble(bParam);
    c = Double.parseDouble(cParam);
    d = Double.parseDouble(dParam);
}catch(NumberFormatException e){
    response.sendError(400, "Дані введені не коректно");
    return;
}

if(c==0){
    response.sendError(400, "Ділення на 0 неможливе");
    return;
}

double y = Math.sqrt(Math.abs(Math.sin(a) - 4 * Math.log(b) / Math.pow(c,
d))) ;

Cookie cookie1 = new Cookie("a1", aParam);
cookie1.setMaxAge(60*60*24*2);
Cookie cookie2 = new Cookie("b1", bParam);
cookie2.setMaxAge(60*60*24*2);
Cookie cookie3 = new Cookie("c1", cParam);
cookie3.setMaxAge(60*60*24*2);
Cookie cookie4 = new Cookie("d1", dParam);
cookie4.setMaxAge(60*60*24*2);
response.addCookie(cookie1);
response.addCookie(cookie2);
response.addCookie(cookie3);
response.addCookie(cookie4);

PrintWriter out = response.getWriter();
out.println("<html><body>");
out.println("<h1>" + "<img alt=\"err\" src=\"images/equation1.png\"/>" + "=" +
y+"</h1>");
out.println("</body></html>");
}

public void destroy() {
}
}

```

Код 1.1 – FormulaServlet_1

```

package com.example.lab_02;

import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.Cookie;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

import java.io.IOException;
import java.io.PrintWriter;

@WebServlet(name = "formulaServlet_2", value = "/formula-2")
public class FormulaServlet_2 extends HttpServlet {

```

```

    public void doGet(HttpServletRequest request, HttpServletResponse response)
throws IOException {response.setContentType("text/html");

    String aParam = request.getParameter("a");
    String bParam = request.getParameter("b");
    String cParam = request.getParameter("c");
    String dParam = request.getParameter("d");

    double a ;
    double b ;
    double c ;
    double d;

    try {
        a = Double.parseDouble(aParam);
        b = Double.parseDouble(bParam);
        c = Double.parseDouble(cParam);
        d = Double.parseDouble(dParam);
    } catch (NumberFormatException e) {
        response.sendError(400, "Дані введені не коректно");
        return;
    }

    if(b==0){
        response.sendError(400, "Ділення на 0 неможливе");
        return;
    }

    double y = ((Math.pow(Math.E,a) + 3 * Math.log10(c)) /
Math.sqrt(Math.pow(b,c))) * Math.abs(Math.atan(d)) ;

    Cookie cookie1 = new Cookie("a2", aParam);
    cookie1.setMaxAge(60*60*24*2);
    Cookie cookie2 = new Cookie("b2", bParam);
    cookie2.setMaxAge(60*60*24*2);
    Cookie cookie3 = new Cookie("c2", cParam);
    cookie3.setMaxAge(60*60*24*2);
    Cookie cookie4 = new Cookie("d2", dParam);
    cookie4.setMaxAge(60*60*24*2);
    response.addCookie(cookie1);
    response.addCookie(cookie2);
    response.addCookie(cookie3);
    response.addCookie(cookie4);

    PrintWriter out = response.getWriter();
    out.println("<html><body>");
    out.println("<h1>" + "<img alt=\"err\" src=\"images/equation2.png\"/>" + "=" +
y+"</h1>");
    out.println("</body></html>");
}

    public void destroy() {
    }
}

```

Код 1.2 – FormulaServlet_2

```

package com.example.lab_02;

import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.Cookie;

```

```

import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

import java.io.IOException;
import java.io.PrintWriter;

@WebServlet(name = "formulaServlet_3", value = "/formula-3")
public class FormulaServlet_3 extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)
throws IOException {response.setContentType("text/html");

        String aParam = request.getParameter("a");
        String bParam = request.getParameter("b");
        String cParam = request.getParameter("c");
        String dParam = request.getParameter("d");

        double a ;
        double b ;
        double c ;
        double d;

        try {
            a = Double.parseDouble(aParam);
            b = Double.parseDouble(bParam);
            c = Double.parseDouble(cParam);
            d = Double.parseDouble(dParam);
        } catch (NumberFormatException e) {
            response.sendError(400, "Дані введені не коректно");
            return;
        }

        double y = Math.pow((2 * Math.sin(a) + Math.cos(Math.abs(b *
Math.sqrt(c))))), d) ;

        Cookie cookie1 = new Cookie("a3", aParam);
        cookie1.setMaxAge(60*60*24*2);
        Cookie cookie2 = new Cookie("b3", bParam);
        cookie2.setMaxAge(60*60*24*2);
        Cookie cookie3 = new Cookie("c3", cParam);
        cookie3.setMaxAge(60*60*24*2);
        Cookie cookie4 = new Cookie("d3", dParam);
        cookie4.setMaxAge(60*60*24*2);
        response.addCookie(cookie1);
        response.addCookie(cookie2);
        response.addCookie(cookie3);
        response.addCookie(cookie4);

        PrintWriter out = response.getWriter();
        out.println("<html><body>");
        out.println("<h1>" + "<img alt=\"err\" src=\"images/equation3.png\"/>" + "=" +
y+"</h1>");
        out.println("</body></html>");
    }

    public void destroy() {
    }
}

```

Код 1.3 – FormulaServlet_3

```

package com.example.lab_02;

import java.io.*;

import jakarta.servlet.http.*;
import jakarta.servlet.annotation.*;

@WebServlet(name = "readCookies", value = "/cookies")
public class ReadCookies extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)
throws IOException {response.setContentType("text/html");

        PrintWriter out = response.getWriter();
        out.println("<html><body>");
        out.println("<h1>Cookies:</h1>");
        for (Cookie c : request.getCookies()) {
            out.println(c.getName() + " " + c.getValue() + "<br/>");
        }
        out.println("</body></html>");
    }

    public void destroy() {
    }
}

```

Код 1.4 – ReadCookies

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Lab 2</title>
</head>
<body>
<p>
<strong>FORMULA 1</strong>
<br>
<form action="formula-1" method="get">
    
    <input type="submit" value=""/>
    <br>
    a=<input type="text" name="a" value=""/>
    <br>
    b=<input type="text" name="b" value=""/>
    <br>
    c=<input type="text" name="c" value=""/>
    <br>
    d=<input type="text" name="d" value=""/>
</form>
<p>
    <strong>FORMULA 2</strong>
<br>
<form action="formula-2" method="get">
    
    <input type="submit" value=""/>
    <br>
    a=<input type="text" name="a" value=""/>
    <br>
    b=<input type="text" name="b" value=""/>
    <br>

```

```

        c=<input type="text" name="c" value=""/>
        <br>
        d=<input type="text" name="d" value=""/>
    </form>
    <p>
        <strong>FORMULA 3</strong>
    <br>
    <form action="formula-3" method="get">
        
        <input type="submit" value="="/>
        <br>
        a=<input type="text" name="a" value=""/>
        <br>
        b=<input type="text" name="b" value=""/>
        <br>
        c=<input type="text" name="c" value=""/>
        <br>
        d=<input type="text" name="d" value=""/>
    </form>
    <p><a href="./vlad-form1">Formula 1</a></p>
    <p><a href="./vlad-form2">Formula 2</a></p>
    <p><a href="./vlad-form3">Formula 3</a></p>
    <br/>
    <p><a href="./cookies">Cookies</a></p>
</body>
</html>

```

Код 1.5 – Реалізація статичної сторінки сайту

```

package com.example.lab_02.vlad;

import jakarta.servlet.ServletException;
import jakarta.servlet.http.Cookie;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

import java.io.IOException;
import java.security.InvalidKeyException;
import java.util.ArrayList;
import java.util.Base64;

public abstract class AbstractVladServlet extends HttpServlet{
    private Cookie _cookie = null;
    protected String _path = "";
    protected String _formFile = "";

    public void init() {

    }

    //Calculate
    public abstract double calculate(double a, double b, double c, double d);

    //Forward request to jsp file
    public void forwardToJspFile(HttpServletRequest request, HttpServletResponse
response) throws IOException, ServletException {
        request.getRequestDispatcher(_formFile).forward(request, response);
    }

    //GET handler
    public void doGet(

```

```

        HttpServletRequest request,
        HttpServletResponse response
    ) throws IOException, ServletException
    {
        try{
            _cookie = CookieHelper.getCookieData(request);
        }catch (InvalidKeyException exc){
            ErrorHandler.setInvalidKeyError(request);
            CookieHelper.deleteCookie(response, _path);
        }

        forwardToJspFile(request, response);
    }

    //POST handler
    public void doPost(
        HttpServletRequest request,
        HttpServletResponse response
    ) throws IOException, ServletException
    {
        String _a = request.getParameter(Defaults._parameterA);
        String _b = request.getParameter(Defaults._parameterB);
        String _c = request.getParameter(Defaults._parameterC);
        String _d = request.getParameter(Defaults._parameterD);

        try{
            _cookie = CookieHelper.getCookieData(request);

            double a = Double.parseDouble(_a);
            double b = Double.parseDouble(_b);
            double c = Double.parseDouble(_c);
            double d = Double.parseDouble(_d);

            double result = calculate(a, b, c, d);

            if (Double.isNaN(result)){
                ErrorHandler.setZeroDivisionError(request);
                forwardToJspFile(request, response);
                return;
            }

            if (Double.isInfinite(result)){
                ErrorHandler.setZeroLogError(request);
                forwardToJspFile(request, response);
                return;
            }

            ArrayList<CalcResult> results =
(ArrayList<CalcResult>)request.getAttribute(Defaults._resultsKey);

            if (results != null && _cookie != null){
                String cookieVal = new
String(Base64.getDecoder().decode(_cookie.getValue()))
                + ResultsHelper.convertToString(false, a, b, c, d, result);

                CalcResult newValue = new CalcResult();
                newValue.a = a;
                newValue.b = b;
                newValue.c = c;
                newValue.d = d;
                newValue.result = result;
                results.add(0, newValue);

                CookieHelper.setCookie(response, cookieVal, _path);
            }
        }
    }
}

```



```

        forwardToJspFile(request, response);
        return;
    }

    String cookieVal = ResultsHelper.convertToString(true, a, b, c, d,
result);

    results = ResultsHelper.parseResults(cookieVal);
    request.setAttribute(Defaults._resultsKey, results);

    CookieHelper.setCookie(response, cookieVal, _path);
} catch (NumberFormatException exc) {
    ErrorHandler.setInvalidRequestDataError(request);
} catch (InvalidKeyException exc) {
    ErrorHandler.setInvalidKeyError(request);
    CookieHelper.deleteCookie(response, _path);
}

    forwardToJspFile(request, response);
}

public void destroy() {
}
}

```

Код 1.6 – AbstractVladServlet

```

package com.example.lab_02.vlad;

public class CalcResult {
    public double a = 0;
    public double b = 0;
    public double c = 0;
    public double d = 0;
    public double result = 0;
}

```

Код 1.7 – CalcResult

```

package com.example.lab_02.vlad;

import jakarta.servlet.http.Cookie;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

import java.security.InvalidKeyException;
import java.util.ArrayList;
import java.util.Base64;

public class CookieHelper {
    //Set results cookie in response
    public static void setCookie(HttpServletResponse response, String value, String
path) {
        Cookie cookie = new Cookie(Defaults._resultsKey,
Base64.getEncoder().encodeToString(value.getBytes()));
        cookie.setMaxAge(60 * 60 * 24 * 2);
        cookie.setHttpOnly(true);
        cookie.setDomain("localhost");
        cookie.setPath(path);
        response.addCookie(cookie);
    }
}

```

```

    }

    //Set results cookie for deletion
    public static void deleteCookie(HttpServletResponse response, String path){
        Cookie deleteCookie = new Cookie(Defaults._resultsKey, "");
        deleteCookie.setMaxAge(0);
        deleteCookie.setHttpOnly(true);
        deleteCookie.setPath(path);
        response.addCookie(deleteCookie);
    }

    //Get results from cookie and set in attribute
    public static Cookie getCookieData(HttpServletRequest request) throws
    InvalidKeyException{
        for (Cookie cookie : request.getCookies()){
            if (cookie.getName().equals(Defaults._resultsKey)){
                String resultsString = new
String(Base64.getDecoder().decode(cookie.getValue().getBytes()));
                ArrayList<CalcResult> results =
ResultsHelper.parseResults(resultsString);
                request.setAttribute(Defaults._resultsKey, results);
                return cookie;
            }
        }
        return null;
    }
}

```

Код 1.8 – CookieHelper

```

package com.example.lab_02.vlad;

public class Defaults {
    public static final String _parameterA = "a";
    public static final String _parameterB = "b";
    public static final String _parameterC = "c";
    public static final String _parameterD = "d";
    public static final String _parameterRes = "res";
    public static final String _resultsKey = "Results";

    public static class Errors{
        public static final String _errorParameter = "Error";
        public static final String _zeroDivisionError = "Can't divide by 0!";
        public static final String _invalidFormData = "Invalid form data!";
        public static final String _invalidKey = "Invalid key detected!";
        public static final String _logFromZero = "Log or negative square root does
not exist!";
    }
}

```

Код 1.9 – Defaults

```

package com.example.lab_02.vlad;

import jakarta.servlet.http.HttpServletRequest;

public class ErrorHandler {
    public static void setZeroDivisionError(HttpServletRequest request){
        request.setAttribute(Defaults.Errors._errorParameter,
Defaults.Errors._zeroDivisionError);
    }
}

```

```

        public static void setInvalidRequestDataError(HttpServletRequest request){
            request.setAttribute(Defaults.Errors._errorParameter,
Defaults.Errors._invalidFormData);
        }

        public static void setInvalidKeyError(HttpServletRequest request){
            request.setAttribute(Defaults.Errors._errorParameter,
Defaults.Errors._invalidKey);
        }

        public static void setZeroLogError(HttpServletRequest request){
            request.setAttribute(Defaults.Errors._errorParameter,
Defaults.Errors._logFromZero);
        }
    }
}

```

Код 1.10 – ErrorHandler

```

package com.example.lab_02.vlad;

import java.security.InvalidKeyException;
import java.util.ArrayList;
import java.util.Collections;

public class ResultsHelper {
    //Parse values into results item string
    public static String convertToString(boolean firstValue, double a, double b,
double c, double d, double result){
        StringBuilder value = new StringBuilder();

        if (!firstValue)
            value.append(";");

        value.append(Defaults._parameterA)
            .append("=")
            .append(a)
            .append(",")
            .append(Defaults._parameterB)
            .append("=")
            .append(b)
            .append(",")
            .append(Defaults._parameterC)
            .append("=")
            .append(c)
            .append(",")
            .append(Defaults._parameterD)
            .append("=")
            .append(d)
            .append(",")
            .append(Defaults._parameterRes)
            .append("=")
            .append(result);

        return value.toString();
    }

    //Parse string to array of results
    public static ArrayList<CalcResult> parseResults(String str) throws
InvalidKeyException {
        ArrayList<CalcResult> result = new ArrayList<>();
    }
}

```

```

        for(String str1 : str.split(";")){
            String[] str2 = str1.split(",");

            CalcResult newResult = new CalcResult();

            for(String str3 : str2){
                String[] keyValue = str3.split("=");
                switch (keyValue[0]){
                    case Defaults._parameterA:{
                        newResult.a = Double.parseDouble(keyValue[1]);
                        break;
                    }
                    case Defaults._parameterB:{
                        newResult.b = Double.parseDouble(keyValue[1]);
                        break;
                    }
                    case Defaults._parameterC:{
                        newResult.c = Double.parseDouble(keyValue[1]);
                        break;
                    }
                    case Defaults._parameterD:{
                        newResult.d = Double.parseDouble(keyValue[1]);
                        break;
                    }
                    case Defaults._parameterRes:{
                        newResult.result = Double.parseDouble(keyValue[1]);
                        break;
                    }
                    default: throw new InvalidKeyException();
                }
            }

            result.add(newResult);
        }

        Collections.reverse(result);
        return result;
    }
}

```

Код 1.11 – ResultsHelper

```

package com.example.lab_02.vlad;

import jakarta.servlet.annotation.WebServlet;

@WebServlet(name = "vladServlet1", value = "/vlad-form1")
public class VladServlet1 extends AbstractVladServlet {

    public VladServlet1() {
        _formFile = "vlad-form1.jsp";
        _path = "/lab_02_war_exploded/vlad-form1";
    }

    @Override
    public double calculate(double a, double b, double c, double d) {
        return Math.sqrt(Math.abs(Math.sin(a) - 4.0 * Math.log(b) / Math.pow(c, d)));
    }
}

```

Код 1.12 – VladServlet1

```

package com.example.lab_02.vlad;

import jakarta.servlet.annotation.WebServlet;

@WebServlet(name = "vladServlet2", value = "/vlad-form2")
public class VladServlet2 extends AbstractVladServlet {
    public VladServlet2(){
        _formFile = "vlad-form2.jsp";
        _path = "/lab_02_war_exploded/vlad-form2";
    }

    @Override
    public double calculate(double a, double b, double c, double d) {
        return (Math.exp(a) + 3 * Math.log10(c) * Math.abs(Math.atan(d))) /
(Math.pow(b, 1 / c));
    }
}

```

Код 1.13 – VladServlet2

```

package com.example.lab_02.vlad;

import jakarta.servlet.annotation.WebServlet;

@WebServlet(name = "vladServlet3", value = "/vlad-form3")
public class VladServlet3 extends AbstractVladServlet {
    public VladServlet3(){
        _formFile = "vlad-form3.jsp";
        _path = "/lab_02_war_exploded/vlad-form3";
    }

    @Override
    public double calculate(double a, double b, double c, double d) {
        return Math.pow(2 * Math.sin(a) + Math.cos(Math.abs(b * Math.sqrt(c))), d);
    }
}

```

Код 1.14 – VladServlet3

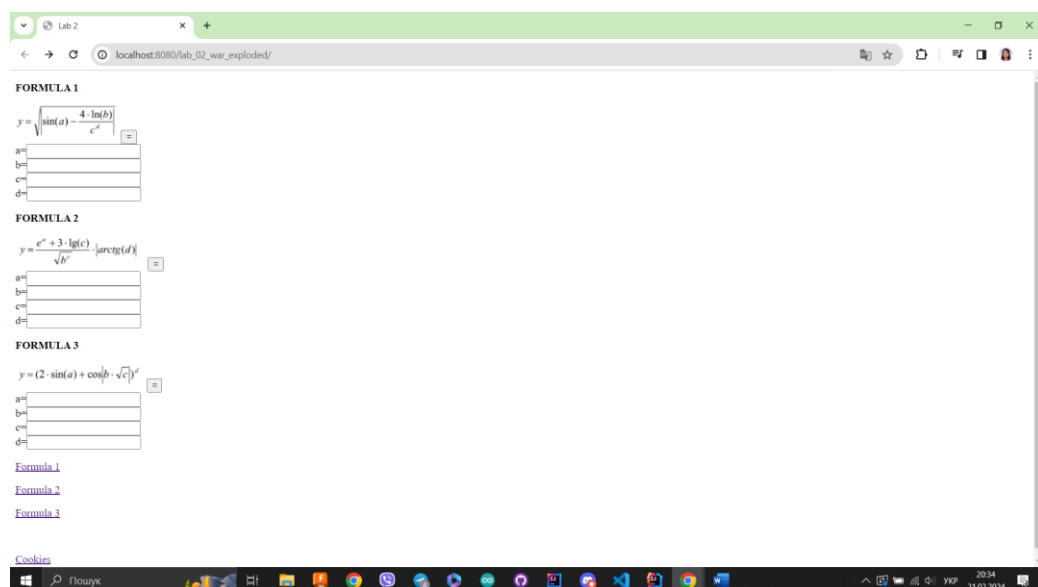


Рисунок 1.1 – Головна сторінка сайту

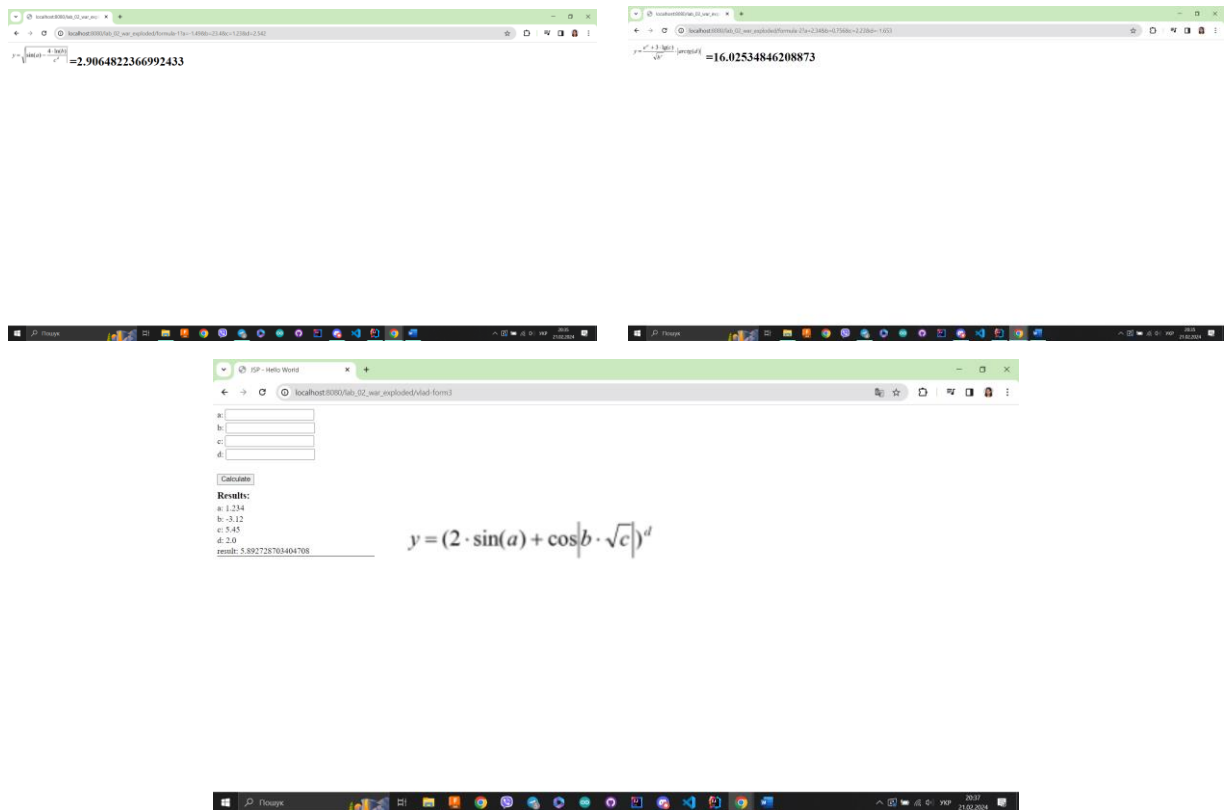


Рисунок 1.2(3, 4) – Приклади інших сторінок сайту

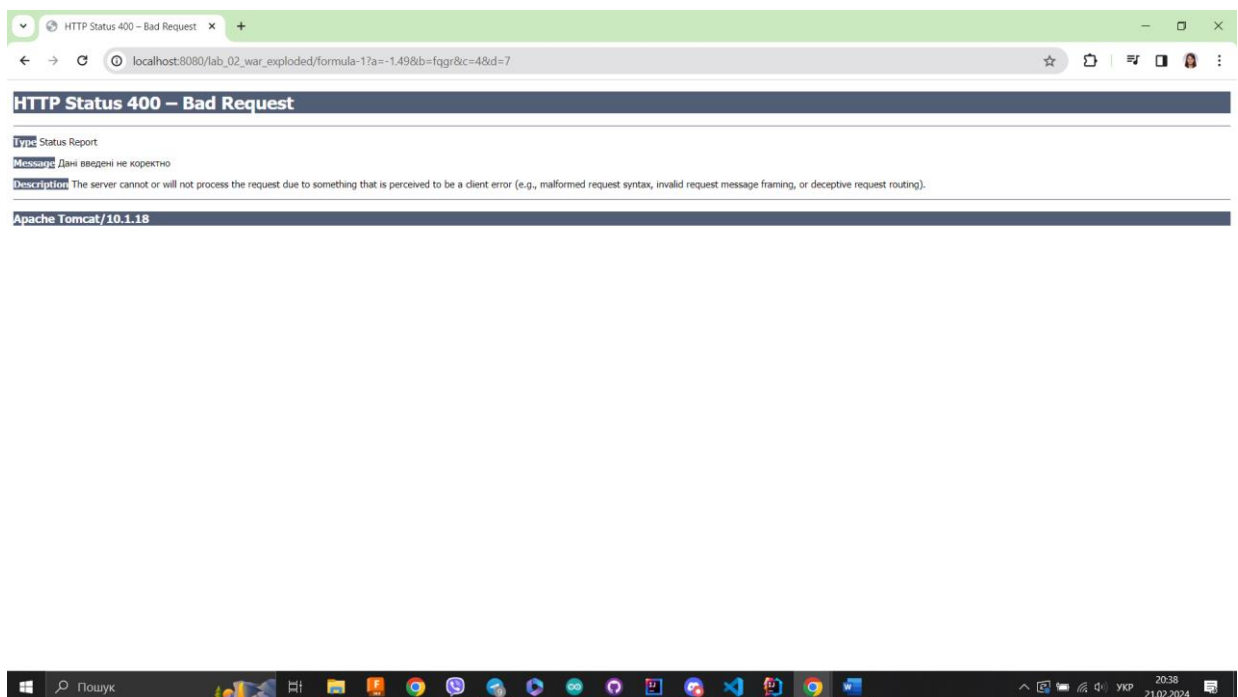


Рисунок 1.5 – Повернення помилки на некоректно введені дані

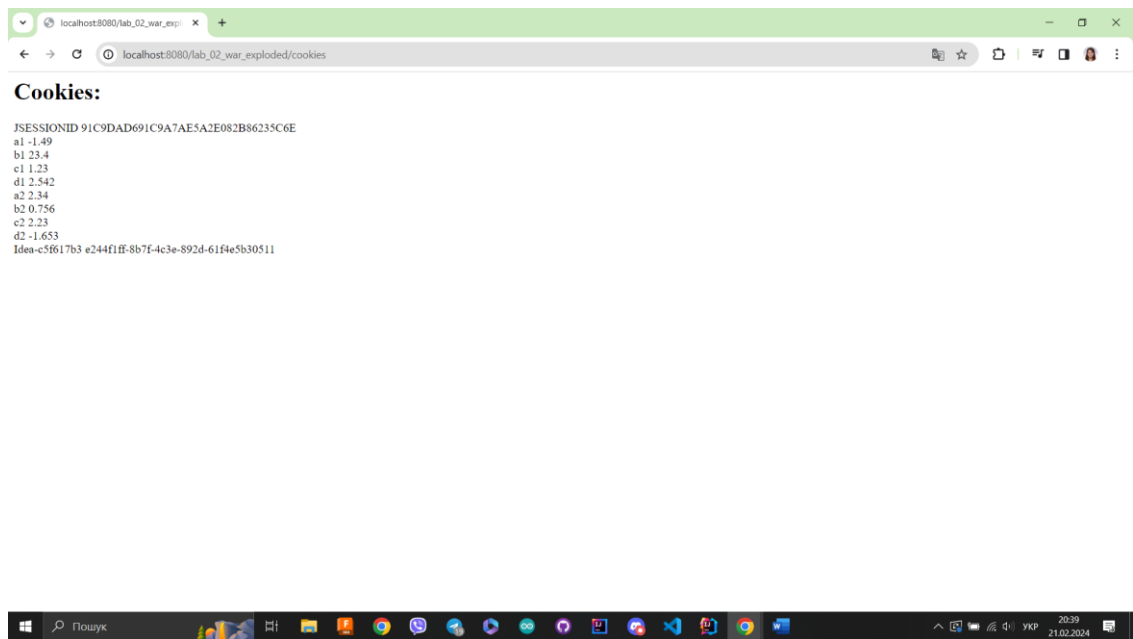


Рисунок 1.6 – Cookies

4. Відповісти на контрольні питання.

Висновки: У результаті виконання лабораторної роботи було успішно опановано основні методи оброблення запитів у сервлетах Java. Ми отримали навички з отримання даних від клієнта, їх оброблення та аналізу, а також виконання дій відповідно до отриманих запитів. Це надасть нам можливість створювати ефективні та надійні веб-додатки, що здатні взаємодіяти з користувачами через веб-інтерфейс.