

# અડવાન્સ જાવા પ્રોગ્રામિંગ (4351603) - શિયાળો 2023 સમાધાન

Milav Dabgar

December 08, 2023

## પ્રશ્ન 1(a) [3 ગુણા]

સ્વિંગ કલાસ હાયરાર્કો દોરો અને સમજાવો.

**જવાબ**

ડાયાગ્રામ:

```
graph TD; Object[Object] --> Component[Component]; Component --> Container[Container]; Container --> JComponent[JComponent]; Container --> Window[Window]; JComponent --> JLabel[JLabel]; JComponent --> JButton[JButton]; JComponent --> Frame[Frame]; JComponent --> Dialog[Dialog]; Window --> JFrame[JFrame]; Window --> JDialog[JDialog];
```

- **Component:** તમામ GUI કોમ્પોનન્ટ્સ માટે બેઝ કલાસ
- **Container:** અન્ય કોમ્પોનન્ટ્સ ધરાવી શકે તેવા કોમ્પોનન્ટ્સ
- **JComponent:** તમામ સ્વિંગ કોમ્પોનન્ટ્સ માટે બેઝ કલાસ

**મેમરી ટ્રીક**

“Objects Contain Components Jointly”

## પ્રશ્ન 1(b) [4 ગુણા]

વિવિધ લેઆઉટ મેનેજરોની યાદી બનાવો. ફલો લેઆઉટ મેનેજરને ઉદાહરણ સાથે સમજાવો.

## જવાબ

લેઆઉટ મેનેજરનું ટેબલ:

લેઆઉટ મેનેજર	વર્ણન
FlowLayout	કોમ્પોન્ટ્સને ડાબેથી જમણે ગોઈવે છે
BorderLayout	પાંચ વિસ્તારો: ઉત્તર, દક્ષિણ, પૂર્વ, પશ્ચિમ, કેન્દ્ર
GridLayout	સમાન કણના લંબચોર્સ ગ્રિડ
CardLayout	કોમ્પોન્ટ્સનો સ્ટેક
BoxLayout	એકલ પંક્તિ અથવા કોલમ

FlowLayout ઉદાહરણ:

```

1 JFrame frame = new JFrame();
2 frame.setLayout(new FlowLayout());
3 frame.add(new JButton("Button1"));
4 frame.add(new JButton("Button2"));
5 frame.setSize(300, 100);
6 frame.setVisible(true);
7

```

- ડિફોલ્ટ એલાઇનમેન્ટ: કોમ્પોન્ટ્સ ડાબેથી જમણે વહે છે
- રેખિંગ: જરૂર પડી તો કોમ્પોન્ટ્સ આગામી લાઇનમાં જાય છે

મેમરી ટ્રીક

"Flow Goes Left Right"

## પ્રશ્ન 1(c) [7 ગુણ]

કાઉન્ટર એપ્લિકેશન માટે જાવા સ્ટિંગ પ્રોગ્રામ વિકસાવો જેમાં લેબલમાં પ્રદર્શિત 0 ની પ્રારંભિક ગણતરી સાથે "વધારો" અને "ઘટાડો" બટન હોય. જ્યારે "વધારો" પર ક્લિક કરવામાં આવે છે, ત્યારે ગણતરી 1 થી વધે છે, અને જ્યારે "ઘટાડો" ક્લિક કરવામાં આવે છે, ત્યારે ગણતરી 1 થી ઓછી થાય છે. જ્યારે કાઉન્ટર 0 થી નીચે જાય ત્યારે message dialog પ્રદર્શિત થવો જોઈએ.

## જવાબ

કોડ:

```

1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.*;
4
5 public class CounterApp extends JFrame implements ActionListener {
6     private int count = 0;
7     private JLabel countLabel;
8     private JButton incButton, decButton;
9
10    public CounterApp() {
11        setTitle("કાઉન્ટર એપ્લિકેશન");
12        setLayout(new FlowLayout());
13
14        countLabel = new JLabel("ગણતરી: " + count);
15        incButton = new JButton("વધારો");
16        decButton = new JButton("ઘટાડો");
17
18        incButton.addActionListener(this);
19        decButton.addActionListener(this);
20

```

```

21     add(countLabel);
22     add(incButton);
23     add(decButton);
24
25     setSize(250, 100);
26     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
27     setVisible(true);
28 }
29
30 public void actionPerformed(ActionEvent e) {
31     if(e.getSource() == incButton) {
32         count++;
33     } else if(e.getSource() == decButton) {
34         count--;
35         if(count < 0) {
36             JOptionPane.showMessageDialog(this, "કાઉન્ટર શૂન્ય થી નીચે!");
37         }
38     }
39     countLabel.setText("ગણતરી: " + count);
40 }
41
42 public static void main(String[] args) {
43     new CounterApp();
44 }
45 }
46

```

- ઇવેન્ટ હેન્ડલિંગ: ActionListener ઇન્ટરફેસ અમલીકરણ
- ડાયલોગ ડિસ્પલે: નેગેટિવ કાઉન્ટર ચેતવણી માટે JOptionPane
- લેબલ અપડેટ: રીએલ-રાઇમ કાઉન્ટ ડિસ્પલે

### મેમરી ટ્રીક

"Increment Decrement Create Dialogs"

OR

### પ્રશ્ન 1(c) [7 ગુણ]

"File" મેનૂમાં મેનૂ આઈટમ્સ "New", "Open" અને "Exit" ધરાવતી સ્વિંગ એપ્લિકેશન બનાવો. જ્યારે વપરાશકર્ત્ત્વ "Exit" ક્લિક કરે છે, ત્યારે એપ્લિકેશન બંધ થવી જોઈએ. ફાઇલ મેનૂ આઈટમ્સ માટે કીબોર્ડ શૉર્ટક્રટ્સ ઉમેરો. "Help" મેનૂમાં મેનૂ આઈટમ "About" પણ ઉમેરો. જ્યારે 'About' ક્લિક કરવામાં આવે ત્યારે એપ્લિકેશન વિશેની માહિતી પ્રદર્શિત કરવા માટે message dialog દેખાવું જોઈએ.

### જવાબ

#### કોડ:

```

1 import javax.swing.*;
2 import java.awt.event.*;
3
4 public class MenuApp extends JFrame implements ActionListener {
5
6     public MenuApp() {
7         setTitle("મેનૂ એપ્લિકેશન");
8
9         JMenuBar menuBar = new JMenuBar();

```

```

10 JMenu fileMenu = new JMenu("File");
11 JMenuItem newItem = new JMenuItem("New");
12 JMenuItem openItem = new JMenuItem("Open");
13 JMenuItem exitItem = new JMenuItem("Exit");
14
15 newItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_N, ActionEvent.CTRL_MASK));
16 openItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_O, ActionEvent.CTRL_MASK));
17 exitItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_X, ActionEvent.CTRL_MASK));
18
19 newItem.addActionListener(this);
20 openItem.addActionListener(this);
21 exitItem.addActionListener(this);
22
23 fileMenu.add(newItem);
24 fileMenu.add(openItem);
25 fileMenu.addSeparator();
26 fileMenu.add(exitItem);
27
28 JMenu helpMenu = new JMenu("Help");
29 JMenuItem aboutItem = new JMenuItem("About");
30 aboutItem.addActionListener(this);
31 helpMenu.add(aboutItem);
32
33 menuBar.add(fileMenu);
34 menuBar.add(helpMenu);
35
36 setJMenuBar(menuBar);
37 setSize(400, 300);
38 setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
39 setVisible(true);
40 }
41
42 public void actionPerformed(ActionEvent e) {
43     String command = e.getActionCommand();
44
45     if(command.equals("Exit")) {
46         System.exit(0);
47     } else if(command.equals("About")) {
48         JOptionPane.showMessageDialog(this,
49             "મેન્યુ એપ્લિકેશન v1.0\શોર્ટકટ્સાં સાથે સ્વચ્છ મેન્યુ દર્શાવે છે");
50     }
51 }
52
53 public static void main(String[] args) {
54     new MenuApp();
55 }
56 }
57 }
58

```

- કીલોડ શોર્ટકટ્સ: Ctrl+N, Ctrl+O, Ctrl+X એક્સેલેટર્સ
- મેન્યુ સ્ક્રાફ્ટ: સૉપેરેટર્સ સાથે File અને Help મેન્યુ
- About ડાયલોગ: પ્રોગ્રામ વર્ણન ડિસ્પલે

### મેમરી ટ્રીક

“Menus Need Shortcuts Always”

## પ્રશ્ન 2(a) [3 ગુણ]

JDBC ડ્રાઇવરના પ્રકારની ચાદી બનાવો. પ્રકાર-4 ડ્રાઇવર સમજાવો.

### જવાબ

#### JDBC ડ્રાઇવરનું ટેબલ:

પ્રકાર	નામ	વર્ણન
Type-1	JDBC-ODBC Bridge	ODBC ડ્રાઇવરનો ઉપયોગ કરે છે
Type-2	Native-API Driver	ડેટાબેસની મૂળ લાઇબ્રેરીઓ વાપરે છે
Type-3	Network Protocol Driver	મિડલવેર સર્વર વાપરે છે
Type-4	Thin Driver	શુદ્ધ જાવા ડ્રાઇવર

#### Type-4 ડ્રાઇવરની વિશેષતાઓ:

- શુદ્ધ જાવા: કોઈ મૂળ કોડની જરૂર નથી
- સીધો સંદેશાચ્ચવહાર: ડેટાબેસ સાથે સીધું જોડાશ
- પ્લેટફોર્મ સ્વતંત્ર: JVM સાથે કોઈપણ OS પર કામ કરે છે

### મેમરી ટ્રીક

“Type Four: Pure Java Door”

## પ્રશ્ન 2(b) [4 ગુણ]

જાવા ફાઉન્ડેશન કલાસ (JFC) ની વિશેષતાઓ સમજાવો.

### જવાબ

#### JFC કોમ્પોનેન્ટ્સ:

- સ્વિંગ: અધ્યતન GUI કોમ્પોનેન્ટ્સ
- AWT: મૂળભૂત GUI ટૂલકિટ
- Accessibility: અક્ષમ વપરાશકર્તાઓ માટે સપોર્ટ
- 2D ગ્રાફિક્સ: વિસ્તૃત ડ્રોઇંગ ક્ષમતાઓ
- Drag and Drop: ફાઇલ ટ્રાન્સફર સપોર્ટ

#### મુખ્ય વિશેષતાઓ:

- Pluggable Look and Feel: UI દેખાવ બદલી શકાય છે
- લાઇટવેઇટ કોમ્પોનેન્ટ્સ: બેહતર કાર્યક્ષમતા
- MVC આર્કિટેક્ચર: પિંતાઓનું વિભાજન
- ઇવેન્ટ હેન્ડલિંગ: મજબૂત ઇવેન્ટ સિસ્ટમ

### મેમરી ટ્રીક

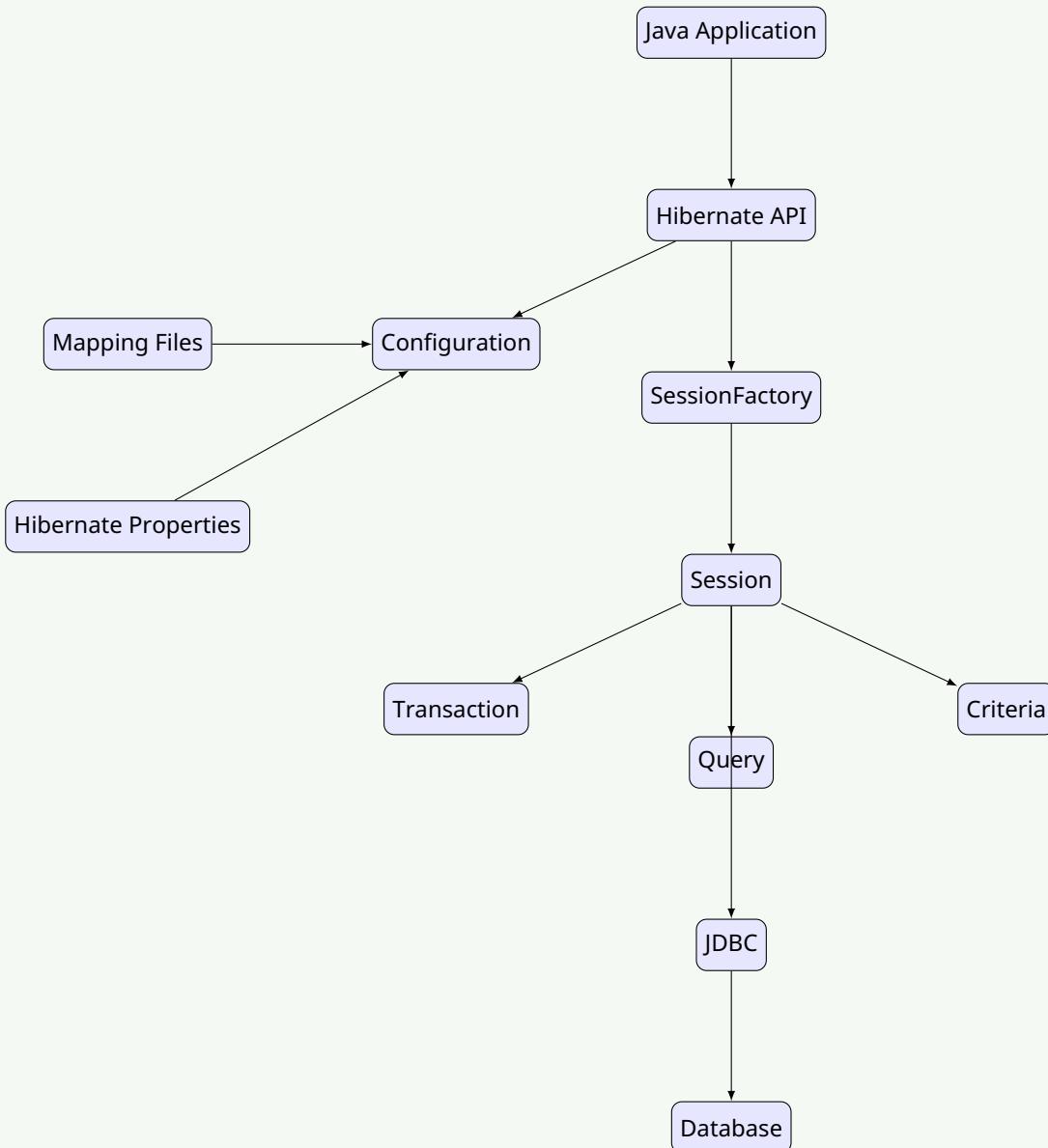
“Java Foundation Creates Swing”

## પ્રશ્ન 2(c) [7 ગુણ]

હાઇબરનેટનું આર્કિટેક્ચર દોરો અને સમજાવો.

### જવાબ

#### ડાયાગ્રામ:



#### આર્કિટેક્ચર કોમ્પોનેન્ટ્સ:

- **Configuration:** મેપિંગ ફાઇલો અને પ્રોપર્ટીઝ વાંચે છે
- **SessionFactory:** Session ઓફજેક્ટ્સ માટે ફેક્ટરી
- **Session:** એલિક્ષન અને ડેટાબેસ વચ્ચે ઇન્ટરફેસ
- **Transaction:** ડેટાબેસ ટ્રાન્ઝેક્શનને દર્શાવે છે
- **Query/Criteria:** ડેટાબેસ કવરીઝ માટે

#### હાઇબ્રનેટના ફાયદા:

- **Object-Relational Mapping:** જાવા ઓફજેક્ટ્સને ડેટાબેસ ટેબલ સાથે મેપ કરે છે
- **અપોઆપ SQL જનરેશન:** મેન્યુઅલ SQL લખવાની જરૂર નથી
- **કેશિંગ:** પ્રથમ-સ્તર અને બીજા-સ્તરની કેશિંગ
- **લેગી લોડિંગ:** જરૂર પડે ત્યારે જ ડેટા લોડ કરે છે

મેમરી ટ્રીક

“Sessions Configure Factories Automatically”

OR

## પ્રશ્ન 2(a) [3 ગુણ]

JDBC API ના ઘટકો સમજાવો.

### જવાબ

#### JDBC API કોષ્ઠોનન્દ્રસ:

- **DriverManager:** ડેટાબેસ ડ્રાઇવર્સનું સંચાલન કરે છે
- **Connection:** ડેટાબેસ કનેક્શનને દર્શાવે છે
- **Statement:** SQL ક્વેરીઓ એક્ઝિક્યુટ કરે છે
- **ResultSet:** ક્વેરી પરિણામો ધરાવે છે
- **SQLException:** SQL એરર્સનું સંચાલન કરે છે

#### કોષ્ઠોનન્દ્રસ:

- ડ્રાઇવર રજિસ્ટ્રેશન: DriverManager.registerDriver()
- કનેક્શન સ્થાપના: DriverManager.getConnection()
- ક્વેરી એક્ઝિક્યુશન: Statement.executeQuery()

### મેમરી ટ્રીક

“Drivers Connect Statements Returning Results”

OR

## પ્રશ્ન 2(b) [4 ગુણ]

કોઈપણ બે સ્વિંગ નિયંત્રણોને ઉદાહરણ સાથે સમજાવો.

### જવાબ

#### JButton કન્ટ્રોલ:

```

1 JButton button = new JButton("મને ફલક્ય કરો");
2 button.addActionListener(new ActionListener() {
3     public void actionPerformed(ActionEvent e) {
4         System.out.println("બટન ફલક્ય થયું!");
5     }
6 });
7 
```

#### JTextField કન્ટ્રોલ:

```

1 JTextField textField = new JTextField(20);
2 textField.setText("અહીં ટેક્સ્ટ લખો");
3 String text = textField.getText();
4 
```

#### વિશેષતાઓ:

- **JButton:** ક્લિક કરવામાં આવે ત્યારે છિયાઓ ટ્રિગર કરે છે
- **JTextField:** સિંગલ-લાઇન ટેક્સ્ટ ઇનપુટ ફીલ્ડ
- ઇવેન્ટ હેન્ડલિંગ: બંને ActionListener સાથે કામ કરે છે

### મેમરી ટ્રીક

“Buttons Text Fields Handle Events”

OR

## પ્રશ્ન 2(c) [7 ગુણ]

Prepared સ્ટેટમેન્ટનો ઉપયોગ કરીને 'info' ડેટાબેઝના 'student' ટેબલમાં એનરોલમેન્ટ\_નંબર, નામ અને ઉંમરનો ડેટા દાખલ કરવા JDBC નો ઉપયોગ કરીને Java પ્રોગ્રામ લખો.

### જવાબ

કોડ:

```

1 import java.sql.*;
2
3 public class StudentInsert {
4     public static void main(String[] args) {
5         String url = "jdbc:mysql://localhost:3306/info";
6         String username = "root";
7         String password = "password";
8
9         try {
10             Class.forName("com.mysql.cj.jdbc.Driver");
11             Connection conn = DriverManager.getConnection(url, username, password);
12
13             String sql = "INSERT INTO student (enrollment_number, name, age) VALUES (?, ?, ?)";
14             PreparedStatement pstmt = conn.prepareStatement(sql);
15
16             pstmt.setString(1, "21IT001");
17             pstmt.setString(2, "જ્યોતિષ્પટેટ");
18             pstmt.setInt(3, 20);
19
20             int rowsAffected = pstmt.executeUpdate();
21             System.out.println("દાખલ થયેલી પંકૃતિઓ: " + rowsAffected);
22
23             pstmt.close();
24             conn.close();
25         } catch(Exception e) {
26             System.out.println("તુટ્ટાં: " + e.getMessage());
27         }
28     }
29 }
30

```

### મુખ્ય કોમ્પોનેન્ટ્સ:

- **PreparedStatement:** SQL injection અટકાવે છે
- પેરામીટર બાઇન્ડિંગ: ? પ્લેસહોલ્ડરનો ઉપયોગ
- કેનેક્શન મેનેજમેન્ટ: યોગ્ય રિસોર્સ કલીનાચાપ
- એક્સોપ્શન હન્ડલિંગ: ડેટાબેસ એરર્સ માટે try-catch

### મેમરી ટ્રીક

“Prepared Statements Prevent Problems”

## પ્રશ્ન 3(a) [3 ગુણ]

સર્વલેટની વિવિધ વિશેષતાઓ સમજાવો.

**જવાબ****સર્વલેટની વિશેષતાઓ:**

- પ્લેટફોર્મ સ્વતંત્ર: જાવા સાથે કોઈપણ OS પર ચાલે છે
- સર્વર-સાઇડ પ્રોસેસિંગ: વેબ સર્વર પર એક્ઝિક્યુટ થાય છે
- પ્રોટોકોલ સ્વતંત્ર: માત્ર HTTP સુધી મયાર્કિટ નથી
- વિસ્તૃત: વિશિષ્ટ જરૂરિયાતો માટે વિસ્તૃત કરી શકાય છે
- મજબૂત: બિલ્ટ-ઇન એક્સોપેશન હેન્ડલિંગ

**વધારાની વિશેષતાઓ:**

- માન્ટિશ્રોડિંગ: એક્સાથે ઘણી વિનંતીઓ હેન્ડલ કરે છે
- પોર્ટબલ: એકવાર લખો, ગમે ત્યાં ચલાયો
- સુરક્ષિત: જાવાની સુરક્ષા વિશેષતાઓ

**મેમરી ટ્રીક****``Servlets Process Protocols Portably''****પ્રશ્ન 3(b) [4 ગુણ]****સર્વલેટની life cycle સમજાવો.****જવાબ****સર્વલેટ લાઇફ સાઇકલ તબક્કાઓ:**

તબક્કો	મેથડ	વર્ણન
લોડિંગ	-	કન્ટેનર દ્વારા સર્વલેટ કલાસ લોડ થાય છે
ઇનસ્ટેન્શિયેશન	-	સર્વલેટ ઓફજેક્ટ બનાવવામાં આવે છે
પ્રારંભિકરણ	init()	સર્વલેટ શરૂ થાય ત્યારે એકવાર કોલ થાય છે
સેવા	service()	દરેક કલાયન્ટ વિનંતી હેન્ડલ કરે છે
વિનાશ	destroy()	સર્વલેટ દૂર કરવા પહેલાં કોલ થાય છે

**લાઇફ સાઇકલ ફ્લો:**

1. કન્ટેનર લોડ કરે છે સર્વલેટ કલાસ
2. ઇનસ્ટન્સ બનાવે છે સર્વલેટનું
3. init() કોલ કરે છે એકવાર
4. service() કોલ કરે છે દરેક વિનંતી માટે
5. destroy() કોલ કરે છે દૂર કરવા પહેલાં

**મેમરી ટ્રીક****``Load Instance Initialize Service Destroy''****પ્રશ્ન 3(c) [7 ગુણ]**

session શું છે? જરૂરી HTML ફાઇલો સહિત HttpSession ઓફજેક્ટનો ઉપયોગ કરીને session manage કેવી રીતે કરી શકાય તે દશાવતો Java servlet પ્રોગ્રામ લખો.

**જવાબ**

**સેશનની વ્યાખ્યા:** સેશન એ બહુવિધ HTTP વિનંતીઓમાં વપરાશકર્તા-વિશિષ્ટ ડેટા સંગ્રહિત કરવાની રીત છે. તે કલાયન્ટ અને સર્વર વચ્ચે સ્થિતિ જાળવે છે.

**સર્વલેટ કોડ:**

```

1 import java.io.*;
2 import javax.servlet.*;
3 import javax.servlet.http.*;
4
5 public class SessionServlet extends HttpServlet {
6     protected void doGet(HttpServletRequest request, HttpServletResponse response)
7         throws ServletException, IOException {
8
9     response.setContentType("text/html; charset=UTF-8");
10    PrintWriter out = response.getWriter();
11
12    HttpSession session = request.getSession(true);
13    String name = request.getParameter("name");
14
15    if(name != null) {
16        session.setAttribute("username", name);
17    }
18
19    String username = (String)session.getAttribute("username");
20    Integer visitCount = (Integer)session.getAttribute("visitCount");
21
22    if(visitCount == null) {
23        visitCount = 1;
24    } else {
25        visitCount++;
26    }
27    session.setAttribute("visitCount", visitCount);
28
29    out.println("<html><body>");
30    out.println("  <h2>સેશન ડેમો</h2>");
31    if(username != null) {
32        out.println("  <p>આવકાર> " + username + "</p>");
33    }
34    out.println("  <p>મુલાકાત> ગણતરી: " + visitCount + "</p>");
35    out.println("  <p>સેશન ID: " + session.getId() + "</p>");
36    out.println("  <a href='index.html'>હોર્ઝુન્ડ્રુ</a> પર પાછા જાઓ</a>");
37    out.println("</body></html>");
38 }
39 }
40

```

### HTML ફાઈલ (index.html):

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>સેશન ડેમો</title>
5     <meta charset="UTF-8">
6 </head>
7 <body>
8     <h2>તમારું નામ દાખલ કરો</h2>
9     <form action="SessionServlet" method="get">
10        નામ: <input type="text" name="name" required>
11        <input type="submit" value="સબમિટ">
12    </form>
13 </body>
14 </html>

```

### સેશન મેનેજમેન્ટ વિશેષતાઓ:

- **getAttribute/setAttribute:** સેશન ડેટા સંગ્રહિત અને પુનઃપ્રાપ્ત કરો
- **સેશન ID:** દરેક સેશન માટે અનન્ય ઓળખકર્તા

- આપોઆપ બનાવત: જરૂર પડે ત્યારે સેશન બનાવવામાં આવે છે

**મેમરી ટ્રીક**

“Sessions Store State Safely”

OR

### પ્રશ્ન 3(a) [3 ગુણ]

servlet માં web.xml ફાઈલ સમજાવો.

#### જવાબ

**web.xml હેતુ:** Web.xml ડિપ્લોયમેન્ટ ડિસ્ટ્રિબ્યુઝન ફાઈલ છે જે સર્વલેટ મેપિંગ, પેરામીટર્સ અને અન્ય વેબ એપ્લિકેશન સેટિંગ્સ કોન્ફિગાર કરે છે. મુખ્ય એલિમેન્ટ્સ:

- servlet:** સર્વલેટ કોન્ફિગરેશન વ્યાખ્યાચિત કરે છે
- servlet-mapping:** URL પેર્ટન સર્વલેટ સાથે મેપ કરે છે
- init-param:** સર્વલેટ પ્રારંભિકરણ પેરામીટર્સ
- welcome-file-list:** ડિફોલ્ટ પૃષ્ઠો

ઉદાહરણ કોન્ફિગરેશન:

```

1 <servlet>
2   <servlet-name>MyServlet</servlet-name>
3   <servlet-class>com.example.MyServlet</servlet-class>
4 </servlet>
5 <servlet-mapping>
6   <servlet-name>MyServlet</servlet-name>
7   <url-pattern>/myservlet</url-pattern>
8 </servlet-mapping>
9

```

**મેમરી ટ્રીક**

“Web XML Maps Servlets”

OR

### પ્રશ્ન 3(b) [4 ગુણ]

સર્વલેટના ફાયદા અને ગેરફાયદા સમજાવો.

#### જવાબ

##### ફાયદા:

- પ્લેટફોર્મ સ્વતંત્ર: જાવા-આધારિત પોર્ટબિલિટી
- પ્રદર્શન: CGI સ્ક્રિપ્ટ્સ કરતાં જડપી
- મજબૂત્ત: એક્સેપ્શન હેન્ડલિંગ અને મેમરી મેનેજમેન્ટ
- સુરક્ષિત: જાવાની સુરક્ષા વિશેષતાઓ
- વિસ્તૃત: વિસ્તૃત અને કસ્ટમાઇઝ કરી શકાય છે

##### ગેરફાયદા:

- જાવા જ્ઞાન જરૂરી: જાવા પ્રોગ્રામિંગ કુશળતાની જરૂર
- પ્રેસ્ન્ટેશન મિશ્રણ: HTML જાવા કોડ સાથે મિક્સ

- ડિબગિંગ જટિલતા: સર્વર-સાઇડ ડિબગિંગ પડકારો
- મયારીદિત ડિજાઇન વિભાજન: તર્ક અને પ્રોગ્રામેશન એક્સાથે

સરખામણી ટેબલ:

પાસું	ફાયદો	ગેરફાયદો
પ્રદર્શન	જડપી એક્ઝિક્યુશન	-
વિકાસ	-	જટિલ ડિબગિંગ
પોર્ટબિલિટી	પ્લેટફોર્મ સ્વતંત્ર	-
કોડ મિશ્રણ	-	જાવામાં HTML

### મેમરી ટ્રીક

“Performance Portability Presents Problems”

OR

## પ્રશ્ન 3(c) [7 ગુણા]

'info' ડેટાબેઝના "student" tableમાંથી ચોક્કસ એન્ટ્રી કાઢી નાખવા માટે જાવા સર્વલેટ પ્રોગ્રામ લખો. સર્વલેટે HTML ફોર્મમાંથી વિદ્યાર્થી ID ઇનપુટ સ્વીકારવું જોઈએ અને ડેટાબેઝમાંથી અનુરૂપ રેકૉર્ડ કાઢી નાખવો જોઈએ.

### જવાબ

સર્વલેટ કોડ:

```

1 import java.io.*;
2 import java.sql.*;
3 import javax.servlet.*;
4 import javax.servlet.http.*;

5
6 public class DeleteStudentServlet extends HttpServlet {
7     protected void doPost(HttpServletRequest request, HttpServletResponse response)
8         throws ServletException, IOException {
9
10     response.setContentType("text/html; charset=UTF-8");
11     PrintWriter out = response.getWriter();
12
13     String studentId = request.getParameter("studentId");
14
15     try {
16         Class.forName("com.mysql.cj.jdbc.Driver");
17         Connection conn = DriverManager.getConnection(
18             "jdbc:mysql://localhost:3306/info", "root", "password");
19
20         String sql = "DELETE FROM student WHERE id = ?";
21         PreparedStatement pstmt = conn.prepareStatement(sql);
22         pstmt.setString(1, studentId);
23
24         int rowsDeleted = pstmt.executeUpdate();
25
26         out.println("<html><body>");
27         out.println("<h2>વિદ્યાર્થી <u> કાઢી નાખવાનું </u> પરિણામ</h2>");
28
29         if(rowsDeleted > 0) {
30             out.println("<p>ID " + studentId + " ધરાવતો વિદ્યાર્થી સફળતાપૂરવક કાઢી નાખવામાં આવ્યો!</p>");
31         } else {
32             out.println("<p>ID " + studentId + " સાથે કોઈ વિદ્યાર્થી મળ્યો નથી</p>");

```

```

33 }
34
35 out.println("<a href='delete.html|બીજો'> વદ્દિયારથી કાઢી નાખો</a>");
36 out.println("</body></html>");
37
38 pstmt.close();
39 conn.close();
40
41 } catch(Exception e) {
42     out.println("<p>તુટે:</p>" + e.getMessage() + "</p>");
43 }
44 }
45 }
46

```

### HTML ફોર્મ (delete.html):

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>વદ્દિયારથી કાઢી નાખો</title>
5     <meta charset="UTF-8">
6 </head>
7 <body>
8     <h2>વદ્દિયારથીનો રેકોર્ડ કાઢી નાખો</h2>
9     <form action="DeleteStudentServlet" method="post">
10        વદ્દિયારથી ID: <input type="text" name="studentId" required>
11        <input type="submit" value="વદ્દિયારથી કાઢી નાખો">
12    </form>
13 </body>
14 </html>
15

```

### મુખ્ય વિશેષતાઓ:

- SQL DELETE ઓપરેશન: ડેટાબેઝમાંથી રેકોર્ડ દૂર કરે છે
- PreparedStatement: SQL injection હુમલાઓ અટકાવે છે
- એર હેન્ડલિંગ: ડેટાબેઝ એક્સેપ્શન-સ માટે try-catch
- વપરાશકર્તા પ્રતિસાદ: સફળતા/નિષ્ફળતા સંદર્ભાઓ

મેમરી ટ્રીક

``Delete Database Data Dynamically''

### પ્રશ્ન 4(a) [3 ગુણ]

JSP અને સર્વલેટ વચ્ચેનો તફાવત સમજાવો.

#### જવાબ

JSP vs સર્વલેટ સરખામણી:

પાસું	JSP	સર્વલેટ
કોડ સ્ટ્રક્ચર	જાવા કોડ સાથે HTML	HTML આઉટપુટ સાથે જાવા
વિકાસ	વેબ ડિઝાઇનરો માટે સરળ	જાવા ડેવલપર્સ માટે બેહતર
કમ્પાઇલેશન	આપોયાપ સર્વલેટમાં કમ્પાઇલ	મેન્યુઅલ કમ્પાઇલેશન જરૂરી
જાળવણી	જાળવવા માટે સરળ	વધુ જટિલ જાળવણી
પ્રદર્શન	પ્રથમ વિનંતી ધીમી	અડ્પી એક્ઝિક્યુશન

મુખ્ય તફાવતો:

- JSP: એમબેદ જાવા સાથે પ્રોજેક્ટેશન-કેન્દ્રિત
- સર્વલેટ: HTML જનરેશન સાથે તર્ક-કેન્દ્રિત
- ઉપયોગ: UI માટે JSP, બિઝનેસ લોજિક માટે સર્વલેટ

મેમરી ટ્રીક

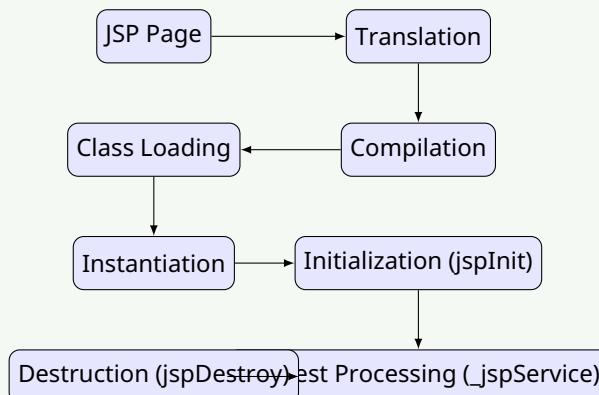
"JSP Presents, Servlets Serve"

## પ્રશ્ન 4(b) [4 ગુણ]

JSPની life cycle સમજાવો.

જવાબ

JSP લાઈફ સાઇકલ તબક્કાઓ:



તબક્કાઓનું વર્ણન:

- ભાષાંતર: JSP સર્વલેટ સોર્સ કોડમાં રૂપાંતરિત થાય છે
- કમ્પાઇલેશન: સર્વલેટ સોર્સ બાઇટકોડમાં કમ્પાઇલ થાય છે
- લોડિંગ: સર્વલેટ કલાસ મેમરીમાં લોડ થાય છે
- ઇન્સ્ટેન્ચિયેશન: સર્વલેટ ઓવ્ઝર્કટ બનાવવામાં આવે છે
- પ્રારંભિકીકરણ: jspInit() મેથ્ડ એકવાર કોલ થાય છે
- સેવા: \_jspService() દરેક વિનંતી હેન્ડ કરે છે
- વિનાશ: દૂર કરવા પહેલાં jspDestroy() કોલ થાય છે

મેમરી ટ્રીક

"Translation Compiles Loading Instances Initialize Service Destroy"

## પ્રશ્ન 4(c) [૭ ગુણ]

એક JSP પ્રોગ્રામ બનાવો જે એક સરળ કેલક્યુલેટર તરીકે કાર્ય કરે. પ્રોગ્રામમાં HTML ફોર્મ હોવું જોઈએ જેમાં બે ટેક્સ્ટબોક્સ નંબરો ઇનપુટ કરવા માટે તથા વપરાશકર્તાઓ ઓપરેશન (ઉમેર, બાદબાકી, ગુણાકાર અથવા ભાગાકાર) પસંદ કરવા માટે ડ્રોપડાઉન મેનુ હોય. જ્યારે વપરાશકર્તા ફોર્મ સબમિટ કરે છે, ત્યારે દાખલ કરેલ નંબરો અને પસંદ કરેલ કામગીરી આગલા પૃષ્ઠ પર મોકલવી જોઈએ. આગલા પૃષ્ઠ પર, વપરાશકર્તાએ પસંદ કરેલા ઓપરેશનના આધારે પરિણામની ગણતરી કરવી જોઈએ અને તેને પ્રદર્શિત કરવી જોઈએ.

### જવાબ

#### HTML ફોર્મ (calculator.html):

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>સરળ કેલક્યુલેટર</title>
5  <meta charset="UTF-8">
6  </head>
7  <body>
8  <h2>સરળ કેલક્યુલેટર</h2>
9  <form action="calculate.jsp" method="post">
10 <table>
11   <tr>
12     <td>પ્રથમ નંબર:</td>
13     <td><input type="number" name="num1" required></td>
14   </tr>
15   <tr>
16     <td>દૂસરો નંબર:</td>
17     <td><input type="number" name="num2" required></td>
18   </tr>
19   <tr>
20     <td>ઓપરેશન:</td>
21     <td>
22       <select name="operation" required>
23         <option value="add">ઉમેર (+)</option>
24         <option value="subtract">બાદબાકી (-)</option>
25         <option value="multiply">ગુણાકાર (*)</option>
26         <option value="divide">ભાગાકાર (/)</option>
27       </select>
28     </td>
29   </tr>
30   <tr>
31     <td colspan="2">
32       <input type="submit" value="ગણતરી કરો">
33       <input type="reset" value="સાફ કરો">
34     </td>
35   </tr>
36 </table>
37 </form>
38 </body>
39 </html>

```

#### JSP કેલક્યુલેટર (calculate.jsp):

```

1  <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
2  <!DOCTYPE html>
3  <html>
4  <head>
5  <title>કેલક્યુલેટર</title>
6  <meta charset="UTF-8">

```

```

7 </head>
8 <body>
9   <h2>કેલક્યુલેટર> પરિણામ</h2>
10
11  <%
12    String num1Str = request.getParameter("num1");
13    String num2Str = request.getParameter("num2");
14    String operation = request.getParameter("operation");
15
16    double num1 = Double.parseDouble(num1Str);
17    double num2 = Double.parseDouble(num2Str);
18    double result = 0;
19    String operationSymbol = "";
20    boolean validOperation = true;
21
22    switch(operation) {
23      case "add":
24        result = num1 + num2;
25        operationSymbol = "+";
26        break;
27      case "subtract":
28        result = num1 - num2;
29        operationSymbol = "-";
30        break;
31      case "multiply":
32        result = num1 * num2;
33        operationSymbol = "*";
34        break;
35      case "divide":
36        if(num2 != 0) {
37          result = num1 / num2;
38          operationSymbol = "/";
39        } else {
40          validOperation = false;
41        }
42        break;
43      default:
44        validOperation = false;
45    }
46  %>
47
48  <div style="border: 1px solid #ccc; padding: 20px; width: 300px;">
49    <h3>ગ્રાહિતરીની> વળિતો:</h3>
50    <p><strong>પ્રથમ> નંબર:</strong> <%= num1 %></p>
51    <p><strong>દ્વારા> નંબર:</strong> <%= num2 %></p>
52    <p><strong>અધોપરેશન>:</strong> <%= operationSymbol %></p>
53
54    <% if(validOperation) { %>
55      <p><strong>પરિણામ:</strong> <%= num1 %> <%= operationSymbol %> <%= num2 %> = <span style="color: blue;
56      &#9675; font-size: 18px;"><%= result %></span></p>
57    <% } else { %>
58      <p style="color: red;"><strong>તુટ્ટાં!</strong> શૂન્ય દ્વારા ભાગાકારની મંજૂરી નથી!</p>
59    <% } %>
60  </div>
61
62  <br />
63  <a href="calculator.html">- કેલક્યુલેટર પર પાછા જાઓ</a>
64 </body>
65 </html>

```

- ફોર્મ વલિડેશન: આવશ્યક ફીલ્ડ્સ અને નંબર ઇનપુટ્સ

- ઓપરેશન પસંદગી: ચાર મૂળભૂત ઓપરેશન્સ સાથે ફ્રોપડાઉન
- એરર હેન્ડલિંગ: શૂન્ય દ્વારા ભાગાકાર અટકાવવું
- યુગ્ર-ફેન્ડલી ડિસ્પ્લે: ફોર્મેટ પરિણામ પ્રસ્તુતિ

મેમરી ટ્રીક

“Calculate Add Subtract Multiply Divide”

OR

## પ્રશ્ન 4(a) [3 ગુણા]

JSP માં પેજ ડાયરેક્ટિવ સમજાવો.

### જવાબ

**પેજ ડાયરેક્ટિવનો હેતુ:** પેજ ડાયરેક્ટિવ JSP કન્ટેનરને પેજ કોન્ફિગરેશન અને પ્રોસેસિંગ વિશે સૂચનાઓ પ્રદાન કરે છે.  
સિન્ક્રેક્સઃ:

```
1 <%@ page attribute="value" %>
2
```

**સામાન્ય એટ્રિબ્યુટ્સ:**

- language:** સ્ક્રિપ્ટિંગ ભાષા (ડિફોલ્ટ: java)
- contentType:** MIME ટાઇપ અને કેરેક્ટર એન્કોડિંગ
- import:** આચાત કરવા માટે જાવા પેકેજુસ
- session:** સેશન સક્ષમ/અક્ષમ (true/false)
- errorPage:** એરર હેન્ડલિંગ પેજ URL

**ઉદાહરણ:**

```
1 <%@ page language="java"
2   contentType="text/html; charset=UTF-8"
3   import="java.util.*;java.sql.*"
4   session="true"
5   errorPage="error.jsp" %>
6
```

મેમરી ટ્રીક

“Page Directives Direct Processing”

OR

## પ્રશ્ન 4(b) [4 ગુણા]

ઉદાહરણ સાથે JSP declaration ટેગ સમજાવો.

### જવાબ

**JSP ડિક્લરેશન ટેગ:** ડિક્લેરેશન ટેગનો ઉપયોગ વેરિએબલ્સ, મેથડ્સ અને કલારીસ ડિક્લેર કરવા માટે થાય છે જે સર્વલેટ કલાસનો ભાગ બને છે.  
સિન્ક્રેક્સઃ:

```
1 <%! declaration code %>
```

2

**ઉદાહરણ:**

```

1 <%!
2   int counter = 0;
3
4   public String getCurrentTime() {
5     return new java.util.Date().toString();
6   }
7
8   private void logVisit() {
9     System.out.println("જુલાકાતા: " + getCurrentTime());
10    }
11  %>
12
13 <html>
14 <body>
15   <h2>કિલેરેશન> ટેગ ડેમો</h2>
16   <%
17     counter++;
18     logVisit();
19   %>
20   <p>જુલાકાત ગણતરી: <%= counter %></p>
21   <p>વર્તમાન> સમય: <%= getCurrentTime() %></p>
22 </body>
23 </html>
24

```

**મુખ્ય મુદ્દાઓ:**

- કલાસ-લેવલ સ્કૉપ: વેરિએબલ્સ ઇન્સ્ટન્સ વેરિએબલ્સ છે
- મેથડ ડિકલેરેશન: મેથડ્સ અને કલાસીસ ડિકલેર કરી શકાય છે
- વિન્ટીઓ વચ્ચે શેર: મૂલ્યો વિન્ટીઓ વચ્ચે ટકી રહે છે
- થ્રેડ સેફ્ટી: સામિલિત એક્સેસ હેન્ડલ કરવાની જરૂર

**મેમરી ટ્રીક****“Declarations Define Class Data”****OR****પ્રશ્ન 4(c) [7 ગુણ]**

કૂકી શું છે? જરૂરી HTML ફાઈલો સહિત કૂકીનો ઉપયોગ કરીને session manage કેવી રીતે કરી શકાય તે દર્શાવતો JSP પ્રોગ્રામ લખો.

**જવાબ**

**કૂકીની વ્યાખ્યા:** કૂકી એ કલાયન્ટ-સાઇડ બ્રાઉઝરમાં સંગૃહિત થતો નાનો ડેટા છે જે HTTP વિન્ટીઓ વચ્ચે સ્થિતિ જાળવવા માટે વપરાય છે.

**HTML ફિર્મ (login.html):**

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>કૂકી સાથે લોગન</title>
5     <meta charset="UTF-8">
6   </head>
7   <body>

```

```

8 <h2>વપરાશકર્તા લોગનિ</h2>
9 <form action="setCookie.jsp" method="post">
10 <table>
11   <tr>
12     <td>વપરાશકર્તા નામ:</td>
13     <td><input type="text" name="username" required></td>
14   </tr>
15   <tr>
16     <td>પાસવર્ડ:</td>
17     <td><input type="password" name="password" required></td>
18   </tr>
19   <tr>
20     <td>મને યાદ રાખો:</td>
21     <td><input type="checkbox" name="remember" value="yes"></td>
22   </tr>
23   <tr>
24     <td colspan="2">
25       <input type="submit" value="લોગનિ">
26     </td>
27   </tr>
28 </table>
29 </form>
30 </body>
31 </html>
32

```

### કૂકી સેટ કરવાનું JSP (setCookie.jsp):

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8" %>
2 <!DOCTYPE html>
3 <html>
4   <head>
5     <title>લોગનિ</title>
6     <meta charset="UTF-8">
7   </head>
8   <body>
9     <%
10       String username = request.getParameter("username");
11       String password = request.getParameter("password");
12       String remember = request.getParameter("remember");
13
14       if("admin".equals(username) && "password".equals(password)) {
15         if("yes".equals(remember)) {
16           Cookie userCookie = new Cookie("username", username);
17           Cookie loginTime = new Cookie("loginTime", String.valueOf(System.currentTimeMillis()));
18
19           userCookie.setMaxAge(7 * 24 * 60 * 60); // 7 દિવસ
20           loginTime.setMaxAge(7 * 24 * 60 * 60);
21
22           response.addCookie(userCookie);
23           response.addCookie(loginTime);
24         }
25       >
26       <h2>લોગનિ!</h2>
27       <p>આવકાર,<%= username %>!</p>
28       <p>લોગનિ! સમય: <%= new java.util.Date() %></p>
29       <a href="welcome.jspસ્વાગત"> પૃષ્ઠ પર જાઓ</a>
30     <%
31       } else {
32     >
33       <h2>લોગનિ! નિષ્ફળ!</h2>
34       <p style="color: red; margin-top: 0;"> વપરાશકર્તા નામ અથવા પાસવર્ડ!</p>

```

```

35     <a href="login.htmlફરી"> પ્રયાસ કરો</a>
36     <%
37     }
38   %>
39 </body>
40 </html>
41

```

### સ્વાગત પૃષ્ઠ JSP (welcome.jsp):

```

1  <%@ page language="java" contentType="text/html; charset=UTF-8" %>
2  <!DOCTYPE html>
3  <html>
4  <head>
5    <title>સ્વાગત> પૃષ્ઠ</title>
6    <meta charset="UTF-8">
7  </head>
8  <body>
9    <h2>સ્વાગત> પૃષ્ઠ</h2>
10   <%
11   Cookie[] cookies = request.getCookies();
12   String savedUsername = null;
13   String loginTime = null;
14
15   if(cookies != null) {
16     for(Cookie cookie : cookies) {
17       if("username".equals(cookie.getName())) {
18         savedUsername = cookie.getValue();
19       } else if("loginTime".equals(cookie.getName())) {
20         loginTime = cookie.getValue();
21       }
22     }
23   }
24
25   if(savedUsername != null) {
26   <%
27     <p>હેલો, <%= savedUsername %>! તમે લોગનિ છો.</p>
28     <% if(loginTime != null) { %>
29       <p>અનુભૂતું> લોગનિ: <%= new java.util.Date(Long.parseLong(loginTime)) %></p>
30     <% } %>
31     <a href="logout.jspલોગઆઉટ"></a>
32   <%
33   } else {
34   <%
35     <p>કૃપા> કરીને આ પૃષ્ઠને એક્સેસ કરવા માટે <a href="login.htmlલોગનિ"></a> કરો.</p>
36   <%
37   }
38   <%>
39 </body>
40 </html>
41

```

- કલાયન-સાઇટ સ્ટોરેજ: બ્રાઉઝરમાં ડેટા સંગ્રહિત થાય છે
- દૃઢતા: બ્રાઉઝર સેશન-સ પછી પણ ટકી શકે છે
- આપોઆપ મોકલવું: દરેક વિનંતી સાથે મોકલવામાં આવે છે
- કદની મર્યાદા: પ્રતિ કૂકી મહત્તમ 4KB

#### મેમરી ટ્રીક

“Cookies Create Client Cache”

## પ્રશ્ન 5(a) [3 ગુણ]

Spring and Spring Boot ની સરખામણી કરો.

### જવાબ

#### Spring vs Spring Boot સરખામણી:

વિશેષતા	Spring Framework	Spring Boot
કોન્ફિગરેશન	XML/Annotation આધારિત	ઓટો-કોન્ફિગરેશન
સેટઅપ સમય	વધુ સમય જરૂરી	ઝડપી સેટઅપ
ડિપેન્ડન્સી મેનેજમેન્ટ	મેન્યુચલ ડિપેન્ડન્સી	સ્ટાર્ટર ડિપેન્ડન્સીઝ
એપ્લોડ સર્વર	બાહ્ય સર્વર જરૂરી	બિલ્દ-ઇન Tomcat/Jetty
પ્રોડક્શન તૈયાર	વધારાનું કોન્ફિગરેશન	તૈયાર-બનેલી વિશેષતાઓ

#### મુખ્ય તફાવતો:

- **Spring Boot:** ડિફોલ્ટ સાથે અભિપ્રાય આધારિત ફેમર્વર્ક
- **Spring Framework:** લવચીક પરંતુ વધુ સેટઅપ જરૂરી
- વિકાસની ઝડપ: Spring Boot વિકસાવવામાં ઝડપી

#### મેમરી ટ્રીક

“Boot Builds Better Beginnings”

## પ્રશ્ન 5(b) [4 ગુણ]

JSP માં તમામ implicit ઓફ્જેક્ટ્સની સૂચિ બનાવો અને કોઈપણ બે સમજાવો.

### જવાબ

#### JSP Implicit ઓફ્જેક્ટ્સની યાદી:

- **request:** HttpServletRequest ઓફ્જેક્ટ
- **response:** HttpServletResponse ઓફ્જેક્ટ
- **session:** HttpSession ઓફ્જેક્ટ
- **application:** ServletContext ઓફ્જેક્ટ
- **out:** JspWriter ઓફ્જેક્ટ
- **page:** વર્તમાન JSP પૃષ્ઠ ઇન્સ્ટન્સ
- **pageContext:** PageContext ઓફ્જેક્ટ
- **config:** ServletConfig ઓફ્જેક્ટ
- **exception:** Exception ઓફ્જેક્ટ (માત્ર એરર પૃષ્ઠો)

#### વિગતવાર સમજૂતી:

##### 1. request ઓફ્જેક્ટ:

```

1 <%
2   String name = request.getParameter("name");
3   String method = request.getMethod();
4   String ip = request.getRemoteAddr();
5 %>
6 <p>નામ: <%= name %></p>
7 <p>મેથ્ડ: <%= method %></p>
8 <p>IP સરનામું: <%= ip %></p>
9

```

##### 2. session ઓફ્જેક્ટ:

```

1 <%
2   session.setAttribute("user", "admin");

```

```

3   String user = (String)session.getAttribute("user");
4   String sessionId = session.getId();
5   %>
6   <pવપરાશકર્તા:> <%= user %></p>
7   <pસેશન> ID: <%= sessionId %></p>
8

```

મેમરી ટ્રીક

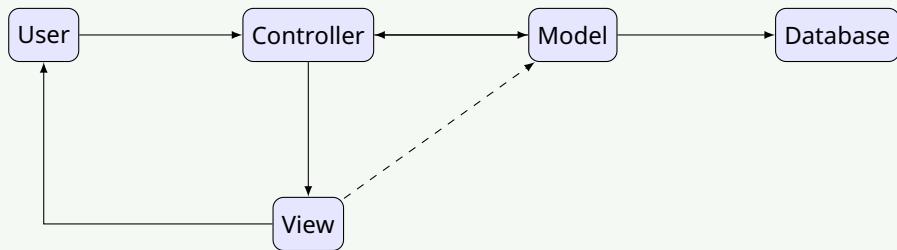
“Request Response Session Application Out”

## પ્રશ્ન 5(c) [7 ગુણ]

MVC આર્કિટેક્ચર સમજાવો.

જવાબ

MVC આર્કિટેક્ચર ડાયાગ્રામ:



MVC કોમ્પોન્ટ્સ:

મોડેલ લેયર:

- ડેટા પ્રતિનિધિત્વ: બિઝનેસ ઓપરેશન્સ અને ડેટા
- બિઝનેસ લોજિક: કોર એપ્લિકેશન કાર્યક્ષમતા
- ડેટાબેસ ઇન્ટરેક્શન: ડેટા એક્સેસ અને મેનેજમેન્ટ
- વેલિડેશન: ડેટા અખંડતા તપાસ

વ્યૂ લેયર:

- પ્રેઝન્ટેશન લોજિક: યુઝર ઇન્ટરફેસ કોમ્પોન્ટ્સ
- ડેટા ડિસ્પ્લે: વપરાશકર્તાને માહિતી બતાવે છે
- યુઝર ઇન્ટરેક્શન: ફોર્મ્સ, બટન્સ, મેન્યુઝ

કન્ટ્રોલર લેયર:

- વિનંતી હન્ડલિંગ: વપરાશકર્તાની વિનંતીઓને પ્રોસેસ કરે છે
- ફ્લો કન્ટ્રોલ: એપ્લિકેશન ફ્લોન્ટુનું સંચાલન કરે છે
- મોડેલ કોઓડિનેશન: મોડેલ લેયર સાથે ઇન્ટરેક્ટ કરે છે
- વ્યૂ પરસંદર્ભ: યોગ્ય વ્યૂ પરસંદ કરે છે

મેમરી ટ્રીક

“Models View Controllers Separate”

OR

## પ્રશ્ન 5(a) [3 ગુણ]

ડિપેન્ડન્સી ઇન્જેક્શન સમજાવો.

## જવાબ

ડિપેન્ડન્સી ઇન્જેક્શનની વ્યાખ્યા: ડિપેન્ડન્સી ઇન્જેક્શન એ ડિજાઇન પેટર્ન છે જ્યાં ઓફ્જેક્ટ પોતે ડિપેન્ડન્સીઝ બનાવવાને બદલે તેઓ પ્રદાન કરવામાં આવે છે.

DI ના પ્રકારો:

- કન્સ્ટ્રક્ટર ઇન્જેક્શન: કન્સ્ટ્રક્ટર દ્વારા ડિપેન્ડન્સીઝ પાસ કરવી
- સેટર ઇન્જેક્શન: સેટર મેથ્ડ્સ દ્વારા ડિપેન્ડન્સીઝ સેટ કરવી
- ફીલ્ડ ઇન્જેક્શન: ફીલ્ડ્સમાં સીધી ડિપેન્ડન્સીઝ ઇન્જેક્ટ કરવી

ઉદાહરણ:

```

1 // DI વનિ
2 public class UserService {
3     private UserRepository repository = new UserRepository();
4 }
5
6 // DI સાથે
7 public class UserService {
8     private UserRepository repository;
9
10    public UserService(UserRepository repository) {
11        this.repository = repository;
12    }
13 }
14

```

DI ના ફાયદા:

- લૂગ કપાલિંગ: કલાસીસ વચ્ચે ઓછી ડિપેન્ડન્સી
- ટેસ્ટેબિલિટી: ડિપેન્ડન્સીઝને મોક કરવાનું સરળ
- લવચીકતા: ઇમ્પ્લિમેન્ટેશન બદલવાનું સરળ

મેમરી ટ્રીક

``Dependencies Injected, Not Instantiated''

OR

## પ્રશ્ન 5(b) [4 ગુણા]

JSTL કોર ટેંસની સૂચિ બનાવો અને ઉદાહરણ સાથે કોઈપણ બે સમજાવો.

## જવાબ

JSTL કોર ટેંસની યાદી:

- c:out: એક્સપ્રેશન વેલ્યુ ડિસ્પ્લે કરે છે
- c:set: વેરિએબલ વેલ્યુ સેટ કરે છે
- c:if: શરતી પ્રોસેસિંગ
- c:choose: બહુવિધ શરતી પ્રોસેસિંગ
- c:forEach: લુપ ઇટરેશન

વિગતવાર ઉદાહરણ:

1. c:forEach ટેંસ:

```

1 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
2
3 <c:set var="numbers" value="1,2,3,4,5" />
4 <ul>
5 <c:forEach var="num" items="${numbers}" varStatus="status">
6     <li>નંબર <${status.index + 1}>: <${num}></li>
7 </c:forEach>

```

```

8 </ul>
9

```

## 2. c:if ટેગ:

```

1 <c:set var="age" value="20" />
2 <c:if test="${age >= 18}">
3   <p style="color: green;*> મતદાન માટે લાયક છો!</p>
4 </c:if>
5 <c:if test="${age < 18}">
6   <p style="color: red;*> મતદાન માટે લાયક નથી!</p>
7 </c:if>
8

```

મેમરી ટ્રીક

“Core Tags Control Conditions”

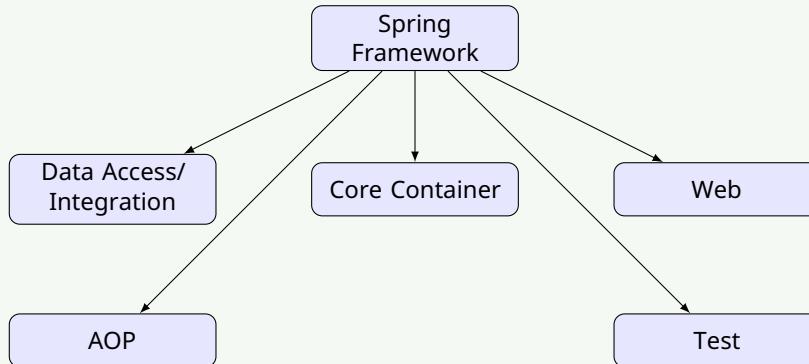
OR

## પ્રક્રિયા 5(c) [7 ગુણ]

સિંગા ફેમવર્કનું આર્કિટેક્ચર સમજાવો.

### જવાબ

#### સિંગા ફેમવર્ક આર્કિટેક્ચર:



#### કોર કન્ટેનર:

- કોર મોડ્યુલ: મૂળભૂત વિશેષતાઓ અને IOC કન્ટેનર
- બીન મોડ્યુલ: બીન ફેક્ટરી અને ડિપેન્ડન્સી ઇન્જેક્શન
- કેટેક્સ્ટ મોડ્યુલ: એપ્લિકેશન કન્ટેક્ટ અને આંતરરાષ્ટ્રીયકરણ
- SpEL મોડ્યુલ: સિંગા એક્સપ્રેશન લેંગ્વેજ

#### ડેટા એક્સેસ/ઇન્ટ્રોફેશન:

- JDBC મોડ્યુલ: ડેટાબેસ કોરિટિવિટી અને ટેમ્પલેટ્સ
- ORM મોડ્યુલ: Hibernate, JPA સાથે ઇન્ટ્રોફેશન
- JMS મોડ્યુલ: જાવા મેસેજ સર્વિસ સપોર્ટ
- ટ્રાન્ઝેક્શન મોડ્યુલ: ડિક્લેરેટિવ ટ્રાન્ઝેક્શન મેનેજમેન્ટ

#### વેબ લંચર:

- વેબ મોડ્યુલ: મૂળભૂત વેબ વિશેષતાઓ અને HTTP યુટિલિટીઝ
- વેબ-MVC મોડ્યુલ: મોડેલ-વ્યૂ-કન્ટ્રોલર ઇમ્પ્લેમેન્ટેશન
- વેબ-સોકેટ મોડ્યુલ: WebSocket સપોર્ટ

#### AOP (Aspect-Oriented Programming):

- કોર્સ-કટિંગ કન્સાર્સ: લોગિંગ, સિક્યુરિટી, ટ્રાન્జેક્શન
- ડિક્વેરેટિવ: એનોટેશન-આધારિત કોન્ફિગરેશન

મેમરી ટ્રીક

“Springs Architecture Supports Complete Applications”