

Subject Name (Gujarati)

4351603 -- Winter 2023

Semester 1 Study Material

Detailed Solutions and Explanations

પ્રશ્ન 1(અ) [3 ગુણ]

સ્વિંગ કલાસ હાયરકોર્ટ દીરો અને સમજાવો.

જવાબ

ડાયગ્રામ:

Mermaid Diagram (Code)

```
{Shaded}
{Highlighting} []
graph LR
    A[Object] --- B[Component]
    B --- C[Container]
    C --- D[JComponent]
    D --- E[JLabel]
    D --- F[JButton]
    D --- G[JTextField]
    D --- H[JTextArea]
    D --- I[JPanel]
    C --- J[Window]
    J --- K[Frame]
    K --- L[JFrame]
    J --- M[Dialog]
    M --- N[JDialog]
{Highlighting}
{Shaded}
```

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

મેમરી ટ્રીક

"ઓફ્જેક્ટ કન્ટેનર કોમ્પોનન્ટ જોઇન"

પ્રશ્ન 1(બ) [4 ગુણ]

વિવિધ લેઆઉટ મેનેજરોની ચારી બનાવો. ફુલો લેઆઉટ મેનેજરને ઉદાહરણ સાથે સમજાવો.

જવાબ

લેઆઉટ મેનેજરનું ટેબલ:

લેઆઉટ મેનેજર	વર્ણન
FlowLayout	કોમ્પોનન્ટ્સને ડાબેથી જમણે ગોઠવે છે
BorderLayout	પાંચ વિસ્તારો: ઉત્તર, દક્ષિણ, પૂર્વ, પશ્ચિમ, કેન્દ્ર
GridLayout	સમાન કદના લંબચોરસ ટ્રિક
CardLayout	કોમ્પોનન્ટ્સનો સ્ટેક

FlowLayout ઉદાહરણ:

```
JFrame frame = new JFrame();
frame.setLayout(new FlowLayout());
frame.add(new JButton("Button1"));
frame.add(new JButton("Button2"));
frame.setSize(300, 100);
frame.setVisible(true);
```

- ડિફોલ્ટ એલાઇનમેન્ટ: કોમ્પોન્ટ્સ ડાબેથી જમણે વહે છે
- રેપિંગ: જરૂર પડે તો કોમ્પોન્ટ્સ આગામી લાઇનમાં જાય છે

મેમરી ટ્રીક

"ફ્લો ડાબે જમણે જાય"

પ્રશ્ન 1(ક) [7 ગુણ]

કાઉન્ટર એપ્લિકેશન માટે જાવા સ્વિંગ 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:**

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class CounterApp extends JFrame implements ActionListener {
    private int count = 0;
    private JLabel countLabel;
    private JButton incButton, decButton;

    public CounterApp() {
        setTitle(" ");
        setLayout(new FlowLayout());

        countLabel = new JLabel(" : " + count);
        incButton = new JButton(" ");
        decButton = new JButton(" ");

        incButton.addActionListener(this);
        decButton.addActionListener(this);

        add(countLabel);
        add(incButton);
        add(decButton);

        setSize(250, 100);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        if(e.getSource() == incButton) {
            count++;
            countLabel.setText(" : " + count);
        } else if(e.getSource() == decButton) {
            count--;
            countLabel.setText(" : " + count);
        }
    }
}
```

```

        count++;
    \} else if(e.getSource() == decButton) \{
        count{--};
        if(count {} 0) \{
            JOptionPane.showMessageDialog(this, "      !");
        \}
        countLabel.setText(" : " + count);
    \}

    public static void main(String[] args) \{
        new CounterApp();
    \}
\}

```

- ઇવેન્ટ હેન્ડલિંગ: ActionListener ઇન્ટરફેસ અમલીકરણ
- ડાયલોગ ડિસ્પ્લે: નેગેટિવ કાઉન્ટ ચેનવણી માટે JOptionPane
- લેબલ અપડેટ: રીએલ-ટાઈમ કાઉન્ટ ડિસ્પ્લે

મેમરી ટ્રીક

“વધારો ઘટાડો ડાયલોગ બનાવો”

પ્રશ્ન 1(ક) અથવા [7 ગુણ]

“File” મેનુમાં મેનુ આઈમ્સ “New”, “Open” અને “Exit” ધરાવતી સ્થિરાંગ એપ્લિકેશન બનાવો. જ્યારે વપરાશકર્તા “Exit” ક્લિક કરે છે, ત્યારે એપ્લિકેશન બંધ થવી જોઈએ. ફાઇલ મેનુ આઈમ્સ માટે કીબોર્ડ શૉટકટ્સ ઉમેરો. “Help” મેનુમાં મેનુ આઈમ “About” પણ ઉમેરો. જ્યારે ‘About’ ક્લિક કરવામાં આવે ત્યારે એપ્લિકેશન વિશેની માહિતી પ્રદર્શિત કરવા માટે message dialog દેખાવું જોઈએ.

જવાબ

કોડ:

```

import javax.swing.*;
import java.awt.event.*;

public class MenuApp extends JFrame implements ActionListener \{

    public MenuApp() \{
        setTitle("      ");
        JMenuBar menuBar = new JMenuBar();

        JMenu fileMenu = new JMenu("File");
        JMenuItem newItem = new JMenuItem("New");
        JMenuItem openItem = new JMenuItem("Open");
        JMenuItem exitItem = new JMenuItem("Exit");

        newItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_N, ActionEvent.CTRL_MASK));
        openItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_O, ActionEvent.CTRL_MASK));
        exitItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_X, ActionEvent.CTRL_MASK));

        newItem.addActionListener(this);
        openItem.addActionListener(this);
        exitItem.addActionListener(this);

        fileMenu.add(newItem);
        fileMenu.add(openItem);
        fileMenu.addSeparator();
        fileMenu.add(exitItem);
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == exitItem) {
            System.exit(0);
        }
    }
}

```

```

JMenu helpMenu = new JMenu("Help");
JMenuItem aboutItem = new JMenuItem("About");
aboutItem.addActionListener(this);
helpMenu.add(aboutItem);

menuBar.add(fileMenu);
menuBar.add(helpMenu);

setJMenuBar(menuBar);
setSize(400, 300);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setVisible(true);
\}

public void actionPerformed(ActionEvent e) \{
String command = e.getActionCommand();

if(command.equals("Exit")) \{
System.exit(0);
\} else if(command.equals("About")) \{
 JOptionPane.showMessageDialog(this,
"           v1.0{n}           ");
\}
\}

public static void main(String[] args) \{
new MenuApp();
\}
\}

```

- કીબોર્ડ શૉર્કટ્સ: Ctrl+N, Ctrl+O, Ctrl+X એક્સેલેરેટર્સ
- મેનુ સ્ક્રૉપ્ટર: સેપેરેટર્સ સાથે File અને Help મેનુ
- About ડાયલોગ: પ્રોગ્રામ વર્ણન ડિસ્પ્લે

મેમરી ટ્રીક

“મેનૂને શૉર્કટની જરૂર હુંમેશા”

પ્રશ્ન 2(અ) [3 ગુણ]

JDBC દ્વારા પ્રકારની યાદી બનાવો. પ્રકાર-4 દ્વારા સમજાવો.

જવાબ

JDBC દ્વારા પ્રકારનું ટેબલ:

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

Type-4 દ્વારા વિશેષતાઓ:

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

મેમરી ટ્રીક

"ટાઇપ ચાર: શુદ્ધ જાવા દ્વાર"

પ્રશ્ન 2(બ) [4 ગુણ]

જાવા ફાઉન્ડેશન કલાસ (JFC) ની વિશેષતાઓ સમજાવો.

જવાબ

JFC કોમ્પોનન્ટ્સ:

- સ્થિરંગ: અધ્યતન GUI કોમ્પોનન્ટ્સ
- 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 કોમ્પોનન્ટ્સ: બેહતર કાર્યક્ષમતા
- MVC આર્કિટેક્ચર: ચિંતાઓનું વિભાજન
- ઇવેન્ટ હન્ડલિંગ: મજબૂત ઇવેન્ટ સિસ્ટમ

મેમરી ટ્રીક

"જાવા ફાઉન્ડેશન સ્થિરંગ બનાવે"

પ્રશ્ન 2(ક) [7 ગુણ]

હાઇબરનેટનું આર્કિટેક્ચર દોરો અને સમજાવો.

જવાબ

ડાયાગ્રામ:

Mermaid Diagram (Code)

```
{Shaded}
{Highlighting} []
graph TD
    A[Java Application] --> B[Hibernate API]
    B --> C[Configuration]
    C --> D[SessionFactory]
    D --> E[Session]
    E --> F[Transaction]
    E --> G[Query]
    E --> H[Criteria]
    I[Mapping Files] --> C
    J[Hibernate Properties] --> C
    E --> K[JDBC]
    K --> L[Database]

{Highlighting}
{Shaded}
```

આર્કિટેક્ચર કોમ્પોનન્ટ્સ:

- Configuration: મેપિંગ ફાઇલો અને પ્રોપર્ટીઝ વાંચે છે
- SessionFactory: Session ઓફ્જેક્ટ્સ માટે ફેક્ટરી
- Session: એપ્લિકેશન અને ડેટાબેસ વચ્ચે ઇન્ટરફેસ
- Transaction: ડેટાબેસ ટ્રાન્ઝેક્શનને દર્શાવે છે

- **Query/Criteria:** ડેટાબેસ કવેરીજ માટે
- હાઇબરનેટના ફાયદા:**
- **Object-Relational Mapping:** જાવા ઓબજેક્ટ્સને ડેટાબેસ ટેબલ સાથે મેપ કરે છે
 - આપોઆપ SQL જનરેશન: મેન્યુઅલ SQL લખવાની જરૂર નથી
 - કેશિંગ: પ્રથમ-સ્તર અને બીજા-સ્તરની કેશિંગ
 - લેઝી લોડિંગ: જરૂર પડે ત્યારે જ ડેટા લોડ કરે છે

મેમરી ટ્રીક

"સેશન કન્ફિગેરેશન ફેક્ટરીઓ આપોઆપ"

પ્રશ્ન 2(અ) અથવા [3 ગુણ]

JDBC API ના ઘટકો સમજાવો.

જવાબ

JDBC API કોમ્પોનેન્ટ્સ:

- **DriverManager:** ડેટાબેસ ડ્રાઇવર્સનું સંચાલન કરે છે
- **Connection:** ડેટાબેસ કનેક્શનનો દર્શાવે છે
- **Statement:** SQL કવેરી એક્ઝિક્યુટ કરે છે
- **ResultSet:** કવેરી પરિણામો ધરાવે છે
- **SQLException:** SQL એરર્સનું સંચાલન કરે છે

કોમ્પોનેન્ટ ફુંક્શન્સ:

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

મેમરી ટ્રીક

"ડ્રાઇવર્સ કનેક્ટ સ્ટેટમેન્ટ રિજાફ્ટ આપે"

પ્રશ્ન 2(બ) અથવા [4 ગુણ]

કોઈપણ બે સ્થિંગ નિયંત્રણોને ઉદાહરણ સાથે સમજાવો.

જવાબ

JButton કન્ટ્રોલ:

```
 JButton button = new JButton("      ");
button.addActionListener(new ActionListener() \{
    public void actionPerformed(ActionEvent e) \{
        System.out.println("      !");
    }
\};}
```

JTextField કન્ટ્રોલ:

```
 JTextField textField = new JTextField(20);
textField.setText("      ");
String text = textField.getText();
```

વિશેષતાઓ:

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

પ્રશ્ન 2(ક) અથવા [7 ગુણ]

Prepared સ્ટેટમેન્ટનો ઉપયોગ કરીને 'info' ડેટાબેઝના 'student' ટેબલમાં એનરોલમેન્ટ_નંબર, નામ અને ઉમરનો ડેટા દાખલ કરવા | JDBC નો ઉપયોગ કરીને Java પ્રોગ્રામ લખો.

જવાબ

કોડ:

```
import java.sql.*;

public class StudentInsert {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/info";
        String username = "root";
        String password = "password";

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection conn = DriverManager.getConnection(url, username, password);

            String sql = "INSERT INTO student (enrollment_number, name, age) VALUES (?, ?, ?)";
            PreparedStatement pstmt = conn.prepareStatement(sql);

            pstmt.setString(1, "21IT001");
            pstmt.setString(2, " ");
            pstmt.setInt(3, 20);

            int rowsAffected = pstmt.executeUpdate();
            System.out.println(" : " + rowsAffected);

            pstmt.close();
            conn.close();
        } catch(Exception e) {
            System.out.println(" : " + e.getMessage());
        }
    }
}
```

મુખ્ય કોમ્પોનેન્ટ્સ:

- **PreparedStatement:** SQL injection અટકાવે છે
- **પેરામીટર બાઇન્ડિંગ:** ? પ્લેસહોલ્ડરનો ઉપયોગ
- **કોન્ક્ષનેન્શન મેનેજમેન્ટ:** થોરાય રિસોર્સ કલીનાચપ
- **એક્સોપ્શન હેન્ડલિંગ:** ડેટાબેસ એરર્સ માટે try-catch

પ્રશ્ન 3(અ) [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 મર્યાદિત નથો
- વિસ્તૃત: વિશિષ્ટ જરૂરિયાતો માટે વિસ્તૃત કરી શકાય છે
- મજબૂત: બિલ્ટ-ઇન એક્સોપ્શન હેન્ડલિંગ

વધારાની વિશેષતાઓ:

- માલ્ટિથ્રેડિંગ: એક્સાથે ધાણી વિનંતીઓ હેન્ડલ કરે છે
- પોર્ટાબલ: એકવાર લખો, ગમે ત્યાં ચલાવો
- સુરક્ષિત: જાવાની સુરક્ષા વિશેષતાઓ

મેમરી ટ્રીક

"સર્વલેટ પ્રોટોકોલ પોર્ટાબલી પ્રોસેસ કરે"

પ્રશ્ન 3(બ) [4 ગુણ]

સર્વલેટની life cycle સમજાવો.

જવાબ

સર્વલેટ લાઇફ સાઇકલ તબક્કાઓ:

તબક્કો	મેથડ	વર્ણન
લોડિંગ	-	કન્નેનર દ્વારા સર્વલેટ કલાસ લોડ થાય છે
ઇનસ્ટેન્શિયેશન	-	સર્વલેટ ઓફજેક્ટ બનાવવામાં આવે છે
પ્રારંભિકીકરણ	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:

- કન્નેનર લોડ કરે છે સર્વલેટ કલાસ
- ઇનસ્ટેન્શિયેશન
- init() કોલ કરે છે એકવાર
- service() કોલ કરે છે દરેક વિનંતી માટે
- destroy() કોલ કરે છે દૂર કરવા પહેલાં

મેમરી ટ્રીક

"લોડ ઇન્સ્ટન્સ પ્રારંભ સેવા વિનાશ"

પ્રશ્ન 3(ક) [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:

```
import java.io.*;
```

```

import javax.servlet.*;
import javax.servlet.http.*;

public class SessionServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html; charset=UTF{-8}");
        PrintWriter out = response.getWriter();

        HttpSession session = request.getSession(true);
        String name = request.getParameter("name");

        if(name != null) {
            session.setAttribute("username", name);
        }

        String username = (String)session.getAttribute("username");
        Integer visitCount = (Integer)session.getAttribute("visitCount");

        if(visitCount == null) {
            visitCount = 1;
        } else {
            visitCount++;
        }
        session.setAttribute("visitCount", visitCount);

        out.println("{htmlbody}");
        out.println("{h2 /h2}");
        if(username != null) {
            out.println("{p " + username + " !{/p}}");
        }
        out.println("{p : " + visitCount + " !{/p}}");
        out.println("{p ID: " + session.getId() + " !{/p}}");
        out.println("{a href=index.html /a}");
        out.println("{/body/html}");
    }
}

```

HTML ફાઈલ (index.html):

```

{!DOCTYPE} html{}
{}html{}
{}head{}
    {}title{}    {}title{}
    {}meta charset="UTF{-8"}{}
{/}head{}
{}body{}
    {}h2{}          {}h2{}
    {}form action="SessionServlet" method="get" {}
        : {}input type="text" name="name" required={}
        {}input type="submit" value=" " {}
    {}form{}
{/}body{}
{/}html{}

```

સેશન મેનેજમેન્ટ વિશેષતાઓ:

- **getAttribute/setAttribute:** સેશન ડેટા સંગ્રહિત અને પુનઃપ્રાપ્ત કરો
- **સેશન ID:** દરેક સેશન માટે અન્ય ઓળખકર્તા
- **આપોઆપ બનાવવટ:** જરૂર પડે ત્યારે સેશન બનાવવામાં આવે છે

ਮੇਮਰੀ ਟ੍ਰੀਕ

“सेशन स्थिति सुरक्षित संग्रहित करे”

પ્રશ્ન 3(અ) અથવા [૩ ગુણ]

servlet मां web.xml ફાઇલ સમજાવો.

ଜ୍ଵାବ

web.xml હેતુ: Web.xml ડિપ્લોયમેન્ટ ડિસ્કૉર્ટ ફાઇલ છે જે સર્વલેટ મેપિંગ, પેરામીટર્સ અને અન્ય વેબ એપ્લિકેશન સોટિંગ્સ કોન્ફિગ્યુર કરે છે.

મુખ્ય એલિમેન્ટ્સ:

- **servlet:** सर्वलेट कोन्फिगरेशन व्याख्यायित करे छे
 - **servlet-mapping:** URL पैटर्न सर्वलेट साथे मेप करे छे
 - **init-param:** सर्वलेट प्रारंभिकरण पेरामीटर्स
 - **welcome-file-list:** डिझोल्ट पृष्ठां

ଓଡାହରଣ କୋନ୍ଫାରେଶନ୍:

```
{}servlet{}
    {}servlet{-name}{MyServlet/}servlet{-name}{}
    {}servlet{-class}{com.example.MyServlet/}servlet{-class}{}
{/servlet{}
{}servlet{-mapping}{}
    {}servlet{-name}{MyServlet/}servlet{-name}{}
    {}url{-pattern}{/myservlet/}url{-pattern}{}
{/servlet{-mapping}{}}
```

ਮੇਮਰੀ ਟ੍ਰੀਕ

“વેબ XML સર્વલેટ મેપ કરે”

પ્રશ્ન 3(બ) અથવા [4 ગુણ]

સર્વલેટ્સના ફાયદા અને ગેરફાયદા સમજાવો.

ଜ୍ଵାବୁ

କ୍ରିୟଦା:

- પ્લેટફોર્મ સ્વતંત્ર: જાવા-આધારિત પોર્ટિબિલિટી
 - પ્રદર્શન: CGI સ્કિપ્ટ્સ કરતાં ઝડપી
 - મજબૂત: એક્સોશન હેન્ડલિંગ અને મેમરી મેનેજમેન્ટ
 - સુરક્ષિત: જાવાની સુરક્ષા વિશેષતાઓ
 - વિસ્તૃત: વિસ્તૃત અને કસ્ટમાઇઝ કરી શકાય છે

ગેરકાયદા:

- **જાવા ક્રાન જરૂરી:** જાવા પ્રોગ્રામિંગ કુશળતાની જરૂર
 - **પ્રોજેક્ટેશન મિશન્સ:** HTML જાવા કોડ સાથે મિક્સ
 - **ડિવાંગ જટિલતા:** સર્વર-સાઇટ ડિવાંગ પડકારો
 - **મધ્યાદિત ડિઝાઇન વિભાગન:** તર્ક અને પ્રોજેક્ટેશન એક્સાયે

સરખામણી ટેબલ:

પાસુ	ફાયદો	ગેરફાયદો
પ્રદર્શન	જડપી એક્ઝિક્યુશન	-
વિકાસ	-	જટિલ ડિબર્જિંગ
પોર્ટબિલિટી	પ્લેટફોર્મ સ્વતંત્ર	-

મેમરી ટ્રીક

“પ્રદર્શન પોર્ટબિલિટી સમસ્યાઓ પ્રસ્તુત કરે”

પ્રશ્ન 3(ક) અથવા [7 ગુણ]

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

જવાબ

સર્વલેટ કોડ:

```

import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class DeleteStudentServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html; charset=UTF{-8}");
        PrintWriter out = response.getWriter();

        String studentId = request.getParameter("studentId");

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection conn = DriverManager.getConnection(
                "jdbc:mysql://localhost:3306/info", "root", "password");

            String sql = "DELETE FROM student WHERE id = ?";
            PreparedStatement pstmt = conn.prepareStatement(sql);
            pstmt.setString(1, studentId);

            int rowsDeleted = pstmt.executeUpdate();

            out.println("{htmlbody}");
            out.println("{h2           /h2}");

            if(rowsDeleted > 0) {
                out.println("{pID " + studentId + "           !{/p}}");
            } else {
                out.println("{pID " + studentId + "           {/p}}");
            }

            out.println("{a href=delete.html           /a}");
            out.println("{/body/html}");

            pstmt.close();
            conn.close();
        } catch(Exception e) {
            out.println("{p : " + e.getMessage() + "{/p}}");
        }
    }
}

```

\}

HTML ફોર્મ (delete.html):

```

{!DOCTYPE html}
{}html{}
{}head{}
    {}title{}          {}
    {}meta charset="UTF{-8"}{}
{/}head{}
{}body{}
    {}h2{}          {}
    {}form action="DeleteStudentServlet" method="post" {}
        ID: {}input type="text" name="studentId" required={}
        {}input type="submit" value="{}"
    {}form{}
{/}body{}
{/}html{}

```

મુખ્ય વિશેષતાઓ:

- SQL DELETE ઓપરેશન: ડેટાબેઝિમાંથી રેકૉર્ડ દૂર કરે છે
- PreparedStatement: SQL injection હુમલાઓ અટકાવે છે
- એરર હેન્ડલિંગ: ડેટાબેઝ એક્સોશાન્સ માટે try-catch
- વપરાશકર્ત્રી પ્રતિસાદ: સફળતા/નિષ્ફળતા સંદેશાચો

મેમરી ટ્રીક

"ડેટાબેઝ ડેટા ડાયનેમિકલી ડિલિટ કરો"

પ્રશ્ન 4(અ) [3 ગુણ]

JSP અને સર્વલેટ વચ્ચેનો તફાવત સમજાવો.

જવાબ

JSP vs સર્વલેટ સરખામણી:

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

મુખ્ય તફાવતો:

- JSP: એમ્બેડેડ જાવા સાથે પ્રોઝેક્ટ-કેન્દ્રિત
- સર્વલેટ: HTML જનરેશન સાથે તર્ક-કેન્દ્રિત
- ઉપયોગ: UI માટે JSP, બિજનેસ લોજિક માટે સર્વલેટ

મેમરી ટ્રીક

"JSP પ્રસ્તુત કરે, સર્વલેટ સેવા આપે"

પ્રશ્ન 4(બ) [4 ગુણ]

JSPની life cycle સમજાવો.

જવાબ

JSP લાઇફ સાઇકલ તબક્કાઓ:

Mermaid Diagram (Code)

```

{Shaded}
{Highlighting} []
graph LR
    A[JSP Page] --> B[Translation]
    B --> C[Compilation]
    C --> D[Class Loading]
    D --> E[Instantiation]
    E --> F[Initialization {--> jspInit}]
    F --> G[Request Processing {--> \_jspService}]
    G --> H[Destruction {--> jspDestroy}]
{Highlighting}
{Shaded}
  
```

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 ઓફ્જેક્ટ બનાવવામાં આવે છે
- પ્રારંભીકરણ: jspInit() મેથ્ડ એકવાર કોલ થાય છે
- સેવા: _jspService() દ્વારા વિનંતી હન્ડલ કરે છે
- વિનાશ: દૂર કરવા પહેલાં jspDestroy() કોલ થાય છે

મેમરી ટ્રીક

“ભાષાંતર કમ્પાઇલ લોડિંગ ઇન્સ્ટન્સ પ્રારંભ સેવા વિનાશ”

પ્રશ્ન 4(ક) [7 ગુણ]

એક JSP પ્રોગ્રામ બનાવો જે એક સરળ કેલ્ક્યુલેટર 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 ફ્રોઝાઉન મેન્યુ 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):

```

!DOCTYPE html
html{}
html{}
head{}
title{} title{}
meta charset="UTF-8" {}
head{}
body{}
h2{} h2{}
form action="calculate.jsp" method="post" {}
table{}
tr{}
td{} :td{}
td{} input type="number" name="num1" required{} td{}
tr{}
  
```

```

    {}tr{}
      {}td{} :{/}td{}
        {}td{}input type="number" name="num2" required{/}td{}
    {}tr{}
    {}tr{}
      {}td{} :{/}td{}
      {}td{}
        {}select name="operation" required={}
          {}option value="add"{} (+){/}option{}
          {}option value="subtract"{} (-){/}option{}
          {}option value="multiply"{} (){/}option{}
          {}option value="divide"{} (){/}option{}
        {}select{}
      {}td{}
    {}tr{}
    {}tr{}
      {}td colspan="2" {}
        {}input type="submit" value=" " {}
        {}input type="reset" value=" " {}
      {}td{}
    {}tr{}
  {}table{}
{}form{}
{}body{}
{}html{}

```

JSP क्रमांक (calculate.jsp):

```

<%@ page language="java" contentType="text/html; charset=UTF{-8}" pageEncoding="UTF{-8"}%>
<!DOCTYPE html>
<html>
<head>
  <title></title>
  <meta charset="UTF{-8}">
</head>
<body>
  <h2></h2>

<%>
  String num1Str = request.getParameter("num1");
  String num2Str = request.getParameter("num2");
  String operation = request.getParameter("operation");

  double num1 = Double.parseDouble(num1Str);
  double num2 = Double.parseDouble(num2Str);
  double result = 0;
  String operationSymbol = "";
  boolean validOperation = true;

  switch(operation) \{
    case "add":
      result = num1 + num2;
      operationSymbol = "+";
      break;
    case "subtract":
      result = num1 {-} num2;
      operationSymbol = "-";
      break;
    case "multiply":
      result = num1 * num2;
      operationSymbol = "";
      break;
  }
</body>
</html>

```

```

        case "divide":
            if(num2 != 0) \{
                result = num1 / num2;
                operationSymbol = "";
            \} else \{
                validOperation = false;
            \}
            break;
        default:
            validOperation = false;
    \}
\%{}

{div} style="border: 1px solid #ccc; padding: 20px; width: 300px;"{}
    {h3      :/h3}
    {pstrong :/strong }{\%=} num1 \%\{}{\/p}
    {pstrong :/strong }{\%=} num2 \%\{}{\/p}
    {pstrong :/strong }{\%=} operationSymbol \%\{}{\/p}

    {\%} if(validOperation) \{ \%
        {pstrong :/strong }{\%=} num1 \%\{} {\%=} operationSymbol \%\{} {\%=} num2 \%\{} = {span} st
    {\%} \} else \{ \%
        {p} style="color: red;"{strong :/strong
    {\%} \} \%
    {/div}

    {br} /{}
    {a} href="calculator.html"{
        /a}
{/body}
{/html}

```

મુખ્ય વિશેષતાઓ:

- ફોર્મ વેલિડેશન: આવશ્યક ફીલ્ડ્સ અને નંબર ઇનપુટ્સ
- ઓપરેશન પરસંદગી: ચાર મૂળભૂત ઓપરેશન્સ સાથે ડ્રોપડાઉન
- એરર હેન્ડલિંગ: શૂન્ય દ્વારા ભાગાકાર અટકાવવું
- યુગર-ફેન્ડલી ડિસ્પ્લે: ફોર્મેટ પરિણામ પ્રસ્તુતિ
- નેવિગેશન: કેલક્યુલેટર ફોર્મ પર પાછા જવાની લિંક

મેમરી ટ્રીક

“ગણતરી ઉમેર બાદ ગુણ ભાગ”

પ્રશ્ન 4(અ) અથવા [૩ ગુણ]

JSP માં પેજ ડાયરેક્ટિવ સમજાવો.

જવાબ

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

{\%@ page} attribute="value" {\%}

સામાન્ય એટ્રિબ્યુટ્સ:

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

ઉદાહરણ:

```
{\%@ page} language="java"
    contentType="text/html; charset=UTF{-8}"
    import="java.util.*,java.sql.*"
    session="true"
    errorPage="error.jsp" \%\{}
```

મેમરી ટ્રીક

“પેજ ડાયરેક્ટિવ પ્રોસેસિંગ નિર્દેશિત કરે”

પ્રશ્ન 4(બ) અથવા [4 ગુણ]

ઉદાહરણ સાથે JSP declaration ટેગ સમજાવો.

જવાબ

JSP ડિકલેરેશન ટેગ: ડિકલેરેશન ટેગનો ઉપયોગ વેરિએબલ્સ, મેથડ્સ અને કલાસીસ ડિકલેર કરવા માટે થાય છે જે સર્વલેટ કલાસનો ભાગ બને છે.

સિન્ક્રેષન:

```
{\%!} declaration code \%\{}
```

ઉદાહરણ:

```
{\%!
    int counter = 0;

    public String getCurrentTime() \{
        return new java.util.Date().toString();
    }

    private void logVisit() \{
        System.out.println("      : " + getCurrentTime());
    }
\%\}

{html}
{body}
    {h2      /h2}
    {\%
        counter++;
        logVisit();
    \%
}
    {p      : }{\%=} counter \%\{}{/p}
    {p      : }{\%=} getCurrentTime() \%\{}{/p}
{/body}
{/html}
```

મુખ્ય મુદ્દાઓ:

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

મેમરી ટ્રીક

“ડિકલેરેશન કલાસ ડેટા વ્યાખ્યાયિત કરે”

પ્રશ્ન 4(ક) અથવા [૭ ગુણ]

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

જવાબ

કૂકીની વ્યાખ્યા: કૂકી એ કલાયન્-સાઇડ બ્રાઉઝરમાં સંગ્રહિત થતો નાનો ડેટા છે જે HTTP વિનંતીઓ વરચે સ્થિત જાળવવા માટે વપરાય છે.

HTML ફિર્મ (login.html):

```
!DOCTYPE html{}  
{}html{}  
{}head{}  
    {}title{}      {}title{}  
    {}meta charset="UTF{-8"}{}  
{}/head{}  
{}body{}  
    {}h2{}      {}h2{}  
    {}form action="setCookie.jsp" method="post"{}  
        {}table{}  
            {}tr{}  
                {}td{}      :{}td{}  
                {}td{}input type="text" name="username" required{/td}{}  
            {}tr{}  
            {}tr{}  
                {}td{}      :{}td{}  
                {}td{}input type="password" name="password" required{/td}{}  
            {}tr{}  
            {}tr{}  
                {}td{}      :{}td{}  
                {}td{}input type="checkbox" name="remember" value="yes"/td{}  
            {}tr{}  
            {}tr{}  
                {}td colspan="2"{}  
                    {}input type="submit" value=" " {}  
                {}td{}  
            {}tr{}  
        {}table{}  
    {}form{}  
{}/body{}  
{}/html{}
```

કૂકી સેટ કરવાનું JSP (setCookie.jsp):

```
@\@ page language="java" contentType="text/html; charset=UTF{-8"} \%{}  
{}!DOCTYPE html{}  
{}html{}  
{}head{}  
    {}title      {}title{}  
    {}meta charset="UTF{-8"}{}  
{}/head{}  
{}body{}  
    {}%{}  
        String username = request.getParameter("username");  
        String password = request.getParameter("password");  
        String remember = request.getParameter("remember");  
  
        if("admin".equals(username) && "password".equals(password)) {  
            if("yes".equals(remember)) {  
                Cookie userCookie = new Cookie("username", username);  
                Cookie loginTime = new Cookie("loginTime", String.valueOf(System.currentTimeMillis()));  
            }  
        }  
    {}%{}
```

```

        userCookie.setMaxAge(7 * 24 * 60 * 60); // 7
        loginTime.setMaxAge(7 * 24 * 60 * 60);

        response.addCookie(userCookie);
        response.addCookie(loginTime);
    \}
\%{%
    {h2      !/h2}
    {p , }{\%={}} username \%{!{/p}}
    {p      : }{\%={}} new java.util.Date() \%{!/p}
    {a} href="welcome.jsp"{{ /a}
\%}
    \} else \{
\%{%
    {h2      !/h2}
    {p} style="color: red;"{{ !/p}
    {a} href="login.html"{{ /a}
\%}
    \}
\%{%
\%{%
{/body}
{/html}

```

સ્વાગત પૃષ્ઠા JSP (welcome.jsp):

```

{\%\@ page} language="java" contentType="text/html; charset=UTF{-8"} \%
{\%!DOCTYPE html{}}
{html}
{head}
    {title      /title}
    {meta} charset="UTF{-8"}{}
{/head}
{body}
    {h2      /h2}
    {\%}
        Cookie[] cookies = request.getCookies();
        String savedUsername = null;
        String loginTime = null;

        if(cookies != null) \{
            for(Cookie cookie : cookies) \{
                if("username".equals(cookie.getName())) \{
                    savedUsername = cookie.getValue();
                \} else if("loginTime".equals(cookie.getName())) \{
                    loginTime = cookie.getValue();
                \}
            \}
        \}

        if(savedUsername != null) \{
\%{%
            {p , }{\%={}} savedUsername \%{!{{ .{/p}
\%} if(loginTime != null) \{ \%
            {p      : }{\%={}} new java.util.Date(Long.parseLong(loginTime)) \%{!/p}
\%} \} \%
            {a} href="logout.jsp"{{ /a}
\%}
            \} else \{
\%{%
                {p {{ href="login.html"{{ /a {{ ./p}
\%}
\%

```

```
\%{}  
{/body}  
{/html}
```

કુકીની વિશેષતાઓ:

- કલાયન્-સાઇડ સ્ટોરેજ: બ્રાઉઝરમાં ડેટા સંગ્રહિત થાય છે
 - દૃઢતા: બ્રાઉઝર સેશન-સ પણી પણ ટકી શકે છે
 - આપોઅપ મોકલવું: દરેક વિનંતી સાથે મોકલવામાં આવે છે
 - કદની મર્યાદા: પ્રતિ ફુફ્કી મહત્તમ 4KB

ਮੇਮਰੀ ਟ੍ਰੀਕ

“કુકીજ કલાયન્ટ કેશ બનાવે”

પ્રશ્ન 5(અ) [૩ ગુણ]

Spring and Spring Boot ની સરખામણી કરો.

ଜ୍ଵାବ

Spring vs Spring Boot સરખામણી:

વિશેષતા	Spring Framework	Spring Boot
કોન્ફિગરેશન	XML/Annotation આધારિત	ઓટો-કોન્ફિગરેશન
સેટયપ સમય	વધુ સમય જરૂરી	જડપી સેટયપ
ડિપેન્નસી મેનેજમેન્ટ	મેન્યુઆલ ડિપેન્નસી	સ્ટાર્ટ ડિપેન્નસીજ
એમ્પ્લેડ સર્વર	બાહ્ય સર્વર જરૂરી	બિલ્ટ-ઇન Tomcat/Jetty
પ્રોડક્શન તૈયાર	વધારાનું કોન્ફિગરેશન	તૈયાર-બનેલી વિશેષતાઓ

ਮੁਖ ਤਫ਼ਾਵਤੀ:

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

ਮੇਮਰੀ ਟੀਕ

“બુટ બેહતર શરૂઆત બનાવે”

પ્રશ્ન 5(બ) [4 ગુણ]

JSP मાં તમામ implicit ઓફ્જલેક્ટની સૂચિ બનાવો અને કોઈપણ બે સમજાવો.

ଜ୍ଵାବ

JSP Implicit ઓફ્જેક્ટ્સની યાદી:

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

વિગતવાર સમજૂતી:

1. request ઓફ્જેક્ટ:

```
{\%}
    String name = request.getParameter("name");
    String method = request.getMethod();
    String ip = request.getRemoteAddr();
\%{}
{p : }{\%={}} name \%{}{/p}
{p : }{\%={}} method \%{}{/p}
{pIP : }{\%={}} ip \%{}{/p}
```

2. session ઓફ્જેક્ટ:

```
{\%}
    session.setAttribute("user", "admin");
    String user = (String)session.getAttribute("user");
    String sessionId = session.getId();
\%{%
{p : }{\%={}} user \%{}{/p}
{p ID: }{\%={}} sessionId \%{}{/p}
```

મેમરી ટ્રીક

"વિનંતી પ્રતિસાદ સેશન એપ્લિકેશન આઉટ"

પ્રશ્ન 5(ક) [7 ગુણ]

MVC આર્કિટેક્ચર સમજાવો.

જવાબ

MVC આર્કિટેક્ચર ડાયાગ્રામ:

Mermaid Diagram (Code)

```
{Shaded}
{Highlighting} []
graph LR
    A[User] --> B[Controller]
    B --> C[Model]
    C --> D[Database]
    C --> B
    B --> E[View]
    E --> A
    E -.> C
{Highlighting}
{Shaded}
```

MVC કોમ્પોનન્ટ્સ:

મોડેલ લેયર:

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

વ્યૂ લેયર:

- પ્રેઝન્ટેશન લોજિક: યુઝર ઇન્ટરફેસ કોમ્પોનન્ટ્સ
- ડેટા ડિસ્પ્લે: વરાશકતાને માહિતી બતાવે છે
- યુઝર ઇન્ટરેક્શન: ફોર્મ્સ, બટન્સ, મેન્યુઓ
- ટેમ્પલેટ્સ: પુનઃઉપયોગ કરી શકાય તેવા UI કોમ્પોનન્ટ્સ

કન્ટ્રોલર લેયર:

- વિનંતી હેન્ડલિંગ: વપરાશકર્તાની વિનંતીઓને પ્રોસેસ કરે છે
- ફલો કન્ટ્રોલ: એપ્લિકેશન ફલોનું સંચાલન કરે છે
- મોડેલ કોઓર્ડિનેશન: મોડેલ લેયર સાથે ઇન્ટરેક્ટ કરે છે
- વ્યૂ પસંદગી: યોગ્ય વ્યૂ પસંદ કરે છે

MVC ના ફાયદા:

- ચિંતાઓનું વિભાજન: રૂપણ જવાબદારી વિભાજન
- જાળવણીક્ષમતા: સુધારવા અને વિસ્તૃત કરવા માટે સરળ
- પુનઃઉપયોગિતા: કોમ્પોનેન્ટ્સનો પુનઃઉપયોગ કરી શકાય છે
- ટેસ્ટેબિલિટી: દરેક લેયરનું સ્વતંત્ર પરીક્ષણ
- સમાંતર વિકાસ: ટીમો એકસાથે કામ કરી શકે છે

ઉદાહરણ ફલો:

1. વપરાશકર્તા ફોર્મ સબમિટ કરે છે (View → Controller)
2. કન્ટ્રોલર ઇનપુટ વેલિડેટ કરે છે
3. કન્ટ્રોલર બિઝનેસ લોજિક માટે મોડેલ કોલ કરે છે
4. મોડેલ ડેટાબેસ સાથે ઇન્ટરેક્ટ કરે છે
5. મોડેલ કન્ટ્રોલરને ડેટા પરત કરે છે
6. વ્યૂ વપરાશકર્તાને પરિણામ દર્શાવે છે

મેમરી ટ્રીક

“મોડેલ વ્યૂ કન્ટ્રોલર અલગ કરે”

પ્રશ્ન 5(અ) અથવા [૩ ગુણ]

ડિપેન્ડ-સી ઇન્જેક્શન સમજાવો.

જવાબ

ડિપેન્ડ-સી ઇન્જેક્શનની વ્યાખ્યા: ડિપેન્ડ-સી ઇન્જેક્શન એ ડિઝાઇન પેટર્ન છે જ્યાં ઓફ્જેક્ટ પોતે ડિપેન્ડ-સીજ બનાવવાને બદલે તેઓ પ્રદાન કરવામાં આવે છે.

DI ના પ્રકારો:

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

ઉદાહરણ:

```
// DI
public class UserService {
    private UserRepository repository = new UserRepository();
}

// DI
public class UserService {
    private UserRepository repository;

    public UserService(UserRepository repository) {
        this.repository = repository;
    }
}
```

DI ના ફાયદા:

- લૂગ કપલિંગ: કલાસીસ વરચે ઓછી ડિપેન્ડ-સી
- ટેસ્ટેબિલિટી: ડિપેન્ડ-સીજને મોક કરવાનું સરળ
- લવચીકતા: ઇમ્પ્લિમેન્ટેશન બદલવાનું સરળ

પ્રશ્ન 5(બ) અથવા [4 ગુણ]

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: વેરિયેબલ વેલ્યુ સેટ કરે છે
- c:if: શરતી પ્રોસેસિંગ
- c:choose: બહુવિધ શરતી પ્રોસેસિંગ
- c:forEach: લૂપ ઇટરેશન
- c:forTokens: ટોકન-આધારિત ઇટરેશન
- c:import: કાન્ટેન ઇનકલુડ કરે છે
- c:url: URL જનરેશન
- c:redirect: રીડાયરેક્ટ રિસ્પોન્સ

વિગતવાર ઉદાહરણો:

1. c:forEach ટેંસ:

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

2. c:if ટેંસ:

```
{c:set} var="age" value="20" /{}
{c:if} test="$\{age == 18\}"({})
    {p} style="color: green;"{!/p}
{/c:if}
{c:if} test="$\{age > 18\}"({})
    {p} style="color: red;"{!/p}
{/c:if}
```

ટેંસની વિશેષતાઓ:

- એક્સપ્રેશન લોગવેજ: EL સિન્ટેક્સ \$[expression] નો ઉપયોગ
- શરતી તર્ક: જાવા if-else સ્ટેટમેન્ટ્સને બદલે છે
- લૂપ પ્રોસેસિંગ: કલેકશન-સ પર ઇટરેટ કરે છે
- સ્વચ્છ વિભાજન: JSP માં જાવા કોડ નથી

પ્રશ્ન 5(ક) અથવા [7 ગુણ]

સ્પ્રેંગ ફેમવર્કનું આર્કિટેક્ચર સમજાવો.

સ્પ્રિંગ ફેમવર્ક આર્કિટેક્ચર:

Mermaid Diagram (Code)

```

{Shaded}
{Highlighting} []
graph TD
    A[Spring Framework] --> B[Core Container]
    A --> C[Data Access/Integration]
    A --> D[Web]
    A --> E[AOP]
    A --> F[Test]

    B --> G[Core]
    B --> H[Beans]
    B --> I[Context]
    B --> J[SpEL]

    C --> K[JDBC]
    C --> L[ORM]
    C --> M[JMS]
    C --> N[Transaction]

    D --> O[Web]
    D --> P[Web{-}MVC]
    D --> Q[Web{-}Socket]
    D --> R[Web{-}Portlet]

{Highlighting}
{Shaded}

```

કોર કન્ટેનર:

- કોર મોડ્યુલ: મૂળભૂત વિશેષતાઓ અને IOC કન્ટેનર
- બી-સ મોડ્યુલ: બીન ફેક્ટરી અને ડિપો-ન-સી ઇન્જેક્શન
- કન્ટેક્ટ મોડ્યુલ: એલિક્શન કન્ટેક્ટ અને આંતરરાષ્ટ્રીયકરણ
- SpEL મોડ્યુલ: સ્પ્રિંગ એક્સપ્રેશન લેંગ્વેજ

ડેટા એક્સેસ/ઇન્ફ્રાશેન:

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

વેબ લેયર:

- વેબ મોડ્યુલ: મૂળભૂત વેબ વિશેષતાઓ અને HTTP યુટિલિટીઓ
- વેબ-MVC મોડ્યુલ: મોડેલ-વ્યૂ-કટ્રોલર ઇમ્પ્લિમેન્ટેશન
- વેબ-સોકેટ મોડ્યુલ: WebSocket સપોર્ટ
- વેબ-પોર્ટલેટ મોડ્યુલ: પોર્ટલેટ એપ્લિકેશાન્સ

AOP (Aspect-Oriented Programming):

- કોસ-કટિંગ કન્સન્સ: લોગિં, સિક્યુરિટી, ટ્રાન્ઝેક્શન
- પ્રોક્સી-આધારિત: મેથડ ઇન્ટર્ફેસ
- ડિક્લેરેટિવ: એનોટેશન-આધારિત કોન્ફિગરેશન

સ્પ્રિંગ ફેમવર્કના ફાયદા:

- લાઇટવેઇટ: ન્યૂનતમ ઓવરહેડ
- નોન-ઇનવેલિબિલ: ફેમવર્ક વિશિષ્ટ કોડની જરૂર નથી
- લૂગલી કપદ: ડિપો-ન-સી ઇન્જેક્શન લૂગ કપલિંગને પ્રોત્સાહન આપે છે
- ડિક્લેરેટિવ: એનોટેશન-સ/XML દ્વારા કોન્ફિગરેશન
- વ્યાપક: સંપૂર્ણ એન્ટરપ્રાઇઝ એપ્લિકેશન ફેમવર્ક

મુખ્ય વિશેષતાઓ:

- IOC કન્ટેનર: ઓફ્જેક્ટ લાઇફસાઇકલનું સંચાલન
- AOP સપોર્ટ: કોસ-કટિંગ કન્સન્સ હેન્ડલિંગ
- ટ્રાન્ઝેક્શન મેનેજમેન્ટ: ડિક્લેરેટિવ ટ્રાન્ઝેક્શન-સ
- MVC ફેમવર્ક: વેબ એપ્લિકેશન ડેવલપમેન્ટ
- ટેસ્ટિંગ સપોર્ટ: વ્યાપક ટેસ્ટિંગ યુટિલિટીઓ

મેમરી ટ્રીક

“સ્પ્રિંગનું આર્કિટેક્ચર સંપૂર્ણ એપ્લિકેશન સપોર્ટ કરે”