

## PROGRAMMING APPLICATION - VİZE GİKMİŞ SORULAR

① Inline CSS = <p style = "color: blue;"> This is a text </p>

② <tr>

<th> A </th>

<th colspan = "2"> B </th>

</tr>

<tr>

<td> C </td>

<td> D </td>

<td rowspan = "2"> E </td>

</tr>

<tr>

<td> F </td>

<td> G </td>

</tr>

A	B	
C	D	E
F	G	

③ Why the post method is more secure than the get method?

Explain with examples as well.

→ Data sent is part of URL in GET method.

→ The parameters are not stored in browser history or in web server logs.

→ We cannot send sensitive information like username in GET method

→ We can send sensitive information like password in Post method

④ Explain what the code below does when it runs.

```
<h:button value="#{a.x3}" outcome="b" />
```

⑤ Explain how do we decide which CSS type should be used in an XHTML application?

Internal Style sheet can be used if one single document has a unique style.

External Style sheet is ideal when the style is applied to many pages.

Inline using the style attribute in HTML elements.

⑥ Write a pattern which fits the pattern rule below.

```
<f:validateRegex pattern="\w([-_.]\w+)*@\w+([-_.]\w+)  
*\.\w+([-_.]\w+)*\w"/>
```

⑦ <h:outputText value=<h:panelGrid><tr><td>A</td></tr>  
</h:panelGrid>" />

Hata alıyorum.

⑨ How Many Days Old Are You?

index.xhtml file

~~h:outputText~~

```
<h:panelGrid columns="2">
    <h:outputText value="Year of Birth"/>
    <h:inputText id="birth"
        requiredMessage="Please enter your birth year"
        value="#{bilgiler.birth-year.str}"
        validatorMessage="Birth year must be equal or fewer than 4 digits">
        <f:validateLength maximum="4"/>
    </h:inputText>
    <h:outputText value="Current year"/>
    <h:inputText id="currntyear"
        requiredMessage="Please enter the current year"
        value="#{bilgiler.current-year.str}"
        validatorMessage="Current year must be of 4 digits">
        <f:validateRegex pattern="\d{4}$"/>
    </h:inputText>
</h:panelGrid>
<h:CommandButton value="Calculate" /><br />
<h:outputText value="Your age in days"/>
<h:outputText value="#{bilgiler.hesapla3}"/>
```

## Exam.java File

```
public String hesapla () {  
    birth_year_int = Integer.valueOf(birth_year_str);  
    current_year_int = Integer.valueOf(current_year_str);  
    days_alive_int = (current_year_int - birth_year_int) * 365;  
    days_alive_str = String.valueOf(days_alive_int);  
    return "index";  
}
```

3

-

⑧ Bayi Satın Alma

index.html file

```
<h:form>  
<b> Bayi Satınalma </b>  
<h:selectOneMenu id="urunler" value="#{prc.urun3}">  
<f:selectItem itemValue="ck" itemLabel="Ceket"/>  
<f:selectItem itemValue="gm" itemLabel="Gomlek"/>  
<f:selectItem itemValue="Pantolon" itemLabel="Pantalon"/>  
Adet : < h:inputText id="adetno" value="#{prc.adet_no3}" /> <br />  
<h:commandButton
```

2016 - Örgün - Bütünlene

③ Which will contain the body of the thread = run()

④ MIN\_PRIORITY and MAX\_PRIORITY = 1 & 10

⑩ select sum(OrderPrice).Customer from Order group by Customer;

① start a thread execution = start()

④ The code executes normally and prints "foo".

- ① Runnable Interface = public void run()
- ② A thread to stop executing = Calling notify() method on an object.
- ③ Missing code to start a thread = X run = new X();  
Thread t = new Thread(run);  
t.start();
- ④ User level threads = They are specific to the OS.
- ⑤ Suitable for the Missing code to start a thread = new SchedDemo("SchedDemo").start();
- ⑥ Cookies were originally designed for = Server-side programming.
- ⑦ CSS style, which of the following should be done = "menu" class must be imported for a hyperlink
- ⑧ Datatable = t
- ⑨ Correct parameter passing = <f:setPropertyActionListener target="#User.operation" value="-" />
- ⑩ ResultSet result = sql.executeQuery(); while(result.next())
- ⑪ SQL command will output the lecturer name of the Physics lesson?  
select lecturers.name from lecturers,lessons where lessons.name = 'Physics' and  
lecturers.id = lessons.lecturer\_id
- ⑫ Select lessons.id, lessons.name, lecturers.name, surname from lecturers,lessons where  
lecturers.id = lessons.lecturer\_id ; = 4x4
- ⑬
- ⑭ Update lectures set surname = 'Adam' where surname = 'Adams' and id in  
(select lecturer\_id from lessons where name = 'Calculus1')
- ⑮ public void attrListener(ActionEvent event){  
nickname = (String)event.getComponent().getAttributes().get("username");  
<f:attribute name="username" value="ABC" />

SQL

```

select count(distinct Country) from Customers;
select * from Customers order by Country;
insert into Customers(CustomerName) values ('Serhat');
update Customers set ContactName = 'Elif' where ContactName = 'Serhat';
delete from Customers where CustomerID = 1;

```

## ① setPropertyActionListener

### xhtml kismi

<h:form>

```
ILK Sayı <h:inputText value="#{islem.ilksayi3}" />
ikinci Sayı <h:inputText value="#{islem.ikincisayi3}" />
<h:commandButton id="topla" value="+" actionListener="#{islem.hesapla3}">
    <f:setPropertyActionListener target="#{islem.islemTipi3}" value="+" />
</h:commandButton>
<h:commandButton id="cikar" value="-" actionListener="#{islem.hesapla3}">
    <f:setPropertyActionListener target="#{islem.islemTipi3}" value="-" />
</h:commandButton>
```

<h:form>

#{islem.ilksayi3} #{islem.islemTipi3} #{islem.ikincisayi3} = #{islem.sonuc3}

### JSF kismi

@ManagedBean

@SessionScoped

public class Islem {

public String islemTipi, ilkSayi, ikinciSayi, sonuc;

//getter-setter method

int a, b, c;

public void hesapla(ActionEvent e) {

a = Integer.parseInt(ilkSayi);

b = Integer.parseInt(ikinciSayi);

if ((e.getComponent().getId().equals("topla")))

c = a + b

else

c = a - b

sonuc = String.valueOf(c);

② selectManyCheckBox / selectOneRadio

xhtml Kismi

<h:form>

```
<h:selectManyCheckBox value="#{userData.values3}">
    <f:selectItem itemLabel="Arts" itemValue="arts"/>
    <f:selectItem itemLabel="Science" itemValue="science"/>
</h:selectManyCheckBox>
<h:commandButton value="Save and List" action="#{userData.show()}" />
<h:outputText value="#{userData.result3}" />
```

</h:form>

JSF kismi

```
@ManagedBean
@SessionScoped
```

```
public class UserData {
```

```
    private List<String> values = new ArrayList<>();
    private String result = " ";
```

```
    public void show() {
```

```
        result = values.toString();
    }
```

```
    public List<String> getValues() { return values; }
```

```
    public void setValues(List<String> values) { this.values = values }
```