



## Spring Framework

**01.a** Mikä on KelloServletin URL, kun asennat ensimmäisen esimerkisovelluksen lokaalille Tomcat-palvelimelle pyörimään?

- ☐ `http://localhost:8080/e01/index.jsp`
- ☐ `http://localhost:8080/e01/kellonaikaservletti`
- ☐ `http://localhost:8080/e01/KelloServlet`

**01.b** Minkälaista muotoilua noudattavan tekstin KelloServlet muodostaa käyttäjän selaimeen?

- ☐ `10.01.2013 9.19`
- ☐ `10.01.2014 09:19:30`
- ☐ `Fri Jan 10 09:19:30 EET 2014`

**02.a** Minkälaisella tagilla jsp-sivulla voidaan ajaa Java-koodia?

- ☐ `<% koodi %>`
- ☐ `<!-- koodi -->`
- ☐ `<%-- koodi --%>`

**02.b** Minkälaisella tagilla jsp-sivulle tulostetaan Java-muuttujan sisältö?

- ☐ `<% muuttuja %>`
- ☐ `<%= muuttuja %>`
- ☐ `<h3>muuttuja</h3>`

**02.c** Minkälaisella SimpleDateFormat-patternilla Date-olio tulostuisi seuraavassa muodossa 2013-01-25?

- ☐ `"yyyy-MM-dd"`
- ☐ `"yyyy-mm-dd"`
- ☐ `"VVVV-KK-PP"`

**03.a** Minkälainen Java-koodirivi kolmannen esimerkkisovelluksen servletissä nappaa http-pyyntöstä lähetetyn tekstikentän sisällön ja tallentaa sen paikalliseen Java-muuttujaan?

- ☐ `PrintWriter wout = response.getWriter();`
- ☐ `wout.println("<!DOCTYPE html>");`
- ☐ `String syotettyNimi = request.getParameter("etunimi");`

**03.b** Minkä niminen form-tagin attribuutti määrittelee osoitteen, johon html-lomake lähetetään?

- ☐ `type`
- ☐ `action`
- ☐ `method`

**03.c** Minkälainen sivu kolmannessa esimerkkisovelluksessa tulee näkyviin (Firefox-selaimella), jos syötät lomakkeeseen seuraavan tekstin: `ETUNIMI<script>alert('SUKUNIMI')</script>`

- ☐ Sivulla lukee Tervehdys ETUNIMI, ja SUKUNIMI aukeaa javascript-popup-ikkunaan.
- ☐ Sivulla lukee Tervehdys ETUNIMI<script>alert('SUKUNIMI')</script>
- ☐ Sivulla lukee Tervehdys ETUNIMI&lt;script&gt;alert('SUKUNIMI')&lt;/script&gt;

**04.a** Minkälaisella koodirivillä neljännessä esimerkkisovelluksessa servletti ohjaa http-pyynnön edelleen jsp-sivulle?

- ☐ `request.setAttribute("henkilo", henk);`
- ☐ `request.getRequestDispatcher("nayta_henkilo.jsp").forward(request, response);`
- ☐ `response.sendRedirect`

**04.b** Mikä `ServletRequest`-rajapintaluokan määrittelemä metodi mahdollistaa servletin tallentaa http-pyynnön ajaksi palvelimen muistiin oliota kuljetettavaksi jsp-sivulle.

- ☐ `public void setAttribute(String nimi, Object arvo)`
- ☐ `public void setParameter(String nimi, Object arvo)`
- ☐ `public void addParameter(String nimi, Object arvo)`

**05.a** Mikä `HttpSession`-rajapintaluokan määrittelemä metodi mahdollistaa olioiden tallentamisen palvelimen muistiin koko session ajaksi?

- ☐ `public void addParameter(String nimi, Object arvo)`
- ☐ `public void setParameter(String nimi, Object arvo)`
- ☐ `public void setAttribute(String nimi, Object arvo)`

**05.b Millä komennolla käyttäjän sessio tuhotaan?**

- ☐ `public void invalidate()`
- ☐ `public void destroy()`
- ☐ `public void removeAttribute(String name)`

**05.c Millä tavalla servlet voi kokonaan uudelleenohjata selaimen toiseen osoitteeseen?**

- ☐ `request.setAttribute(String url, Timestamp when);`
- ☐ `response.sendRedirect(String osoite);`
- ☐ `request.forward(String osoite);`

**06.a Minkälaisella koodirivillä kuudennessa esimerkksiovelluksessa ladataan mysql:n jdbc-ajuri?**

- ☐ `Statement haku = yhteys.createStatement();`
- ☐ `yhteys = DriverManager.getConnection(url, username, password);`
- ☐ `Class.forName("org.mariadb.jdbc.Driver").newInstance();`

**06.b Miten tietokantahaun tulosten läpikäynnissä siirrytään seuraavaan tietueeseen?**

- ☐ Statement-olion `next()`-metodilla
- ☐ ResultSet-olion `next()`-metodilla
- ☐ Driver-luokan `forward()`-kutsulla

**06.c Kuinka tietokantayhteys suljetaan java-ohjelmassa?**

- ☐ Driver-luokan `destroy()`-metodia kutsumalla
- ☐ Connection-olion `close()`-metodia kutsumalla
- ☐ Statement-luokan `close()`-metodia kutsumalla

**07.a Minkä niminen luokka on seitsemännessä esimerkksiovelluksessa vastuussa henkilöiden hakemisesta tietokannasta?**

- ☐ `db_connection.properties`
- ☐ `Henkilo`
- ☐ `HenkiloDAO`

**07.b Mikä toimii avaimen ja arvon erotinmerkinä Javan properties-tiedostoissa?**

- ☐ =
- ☐ -
- ☐ :

**07.c Mikä metodi tarjoilee kutsujalle DBConnectionProperties-luokan ainoaa ilmentymää?**

- ☐ DBConnectionProperties-luokan julkinen staattinen getInstance()-metodi
- ☐ DBConnectionProperties-luokan yksityinen getInstance()-metodi
- ☐ db\_connection.properties-luokan getProperty()-metodi

**07.d Mitä sanalla henk tarkoitetaan sivun henkilot.jsp for-each -silmukassa?**

- ☐ Henkilo-luokan ilmentymistä koostuvaa kokoelmaa (ArrayList)
- ☐ forEach-silmukan yksittäisellä kierroksella käsiteltävää Henkilo-oliota.
- ☐ Vakiota, joka on HashtableEncodingNullableKey-luokan ilmentymä

**08.a Mistä arvosta prepared statementin kysymysmerkkien numerointi aloitetaan?**

- ☐ 0
- ☐ 1
- ☐ -1

**08.b Löytyykö kahdeksannesta esimerkisovelluksesta SQL Injection -haavoittuvuus?**

- ☐ Kanada
- ☐ Kyllä
- ☐ Ei

**08.c Minkä niminen Exceptionista periytyvä luokka kahdeksanteen esimerkisovellukseen on ohjelmoitu?**

- ☐ DBConnectionPropertiesException
- ☐ DAOException
- ☐ DAOPoikkeus

**09.a Mitä vaihtoehtoisia salausalgoritmeja luokka Salaaja tarjoaa käytettäväksi?**

- ☐ MD5, SHA-1, SHA-256, SHA-384, SHA-512
- ☐ MD5, ROT-13
- ☐ CRC, Blowfish

**09.b Montako kertaa yhdeksäs esimerkisovellus pyöryttää salausalgoritmin ympäri yhdelle salasanalle?**

- ☐ 1
- ☐ 100
- ☐ 1000

**09.c Onko suola tallennettu tietokantaan kryptattuna vai selkokiekisenä?**

- ☐ Kryptattuna
- ☐ Selkokiekisenä
- ☐ Kanada

**09.d Miksi WebUserDAO:n lisää-metodi suorittaa tietokantahaun ennen lisäysoperaatiota?**

- ☐ Koska tietokantayhteys pitää aina muistaa sulkea
- ☐ Tarkistetaan, ettei usernamella jo löydy käyttäjää
- ☐ Jotta voidaan varmistaa, että yhteys on auki

**09.e Miksi jsp-sivut halutaan säilyttää WEB-INF -kansion sisällä?**

- ☐ Jotta selaimella ei pysty kutsumaan niitä suoraan (kulkematta servletin kautta)
- ☐ Jotta mobiililaitteetkin löytäisivät ne
- ☐ Jotta hakukoneet osaisivat indeksoida ne

**10.a Kuinka SiteServlet tietää milloin käyttäjä on ohjattava sisäänkirjautumissivulle?**

- ☐ Mikäli käyttäjän tiedot on tallennettu sessiomuuttujaan, käyttäjä on tunnistettu. Muutoin hän on tuntematon käyttäjä, ja hänet ohjataan sisäänkirjautumissivulle.
- ☐ Http-protokollan head-osion user-agent-kenttään kirjattu selaintieto määrää sisäänkirjautumissivun
- ☐ SiteServlet suorittaa tietokantahaun ja tarkistaa löytyykö käyttäjän tietoja kannasta. Mikäli käyttäjää ei löydy, hänet viedään sisäänkirjautumissivulle.

**10.b Mitä uloskirjautuminen tekee sessiolle?**

- ☐ Tallentaa session tietokantaan
- ☐ Tuhoaa session ja sen sisältämät tiedot
- ☐ Tallentaa session requestiin

**10.c Minkä nimiseen sessiomuuttujaan käyttäjän tiedot tallennetaan sisäänkirjautumisen jälkeen kymmenennessä esimerkisovelluksessa?**

- ☐ SESSION\_WEBUSER
- ☐ henk
- ☐ kayttajatiedot

**10.d Minkä luokan missä metodissa sovellus tekee päätöksen siitä, onko annettu salasana oikein vai väärin?**

- ☐ Salaja-luokan salaa()-metodissa
- ☐ SisaankirjautumisServlet-luokan konstruktorissa
- ☐ WebUser-luokan vertaaSalasanaa()-metodissa

**11.a Minkä nimisellä annotaatiolla metodin voi määritellä JUnit-testitapaukseksi?**

- ☐ @Test
- ☐ @TestCase
- ☐ @Case

**11.b Mitä tarkoittaa import static?**

- ☐ Mikäli jossain luokassa kirjoitetaan lauseke import static, avautuu static-pakkauksen kaikki luokat käyttöön.
- ☐ Ulkopuolisen luokan staattiset jäsenet saadaan käyttöön, kuin ne olisivat luokan omia staattisia jäseniä.
- ☐ Import static muuttaa Java-kielen funktionaaliseksi.

**11.c Miksi mock-olioita tarvitaan yksikkötesteihin?**

- ☐ Mock-oliota periyttämällä voidaan luoda automaattisesti testitapauksia. Nämä testitapaukset ovat automaattisesti vaatimusmäärittelyn mukaisia.
- ☐ Mock-olio voidaan asettaa oikean olion paikalle testin ajaksi. Näin yksikkötesti voidaan rajata tutkimaan yhtä toiminnallisuutta kerrallaan.
- ☐ JUnitin testitapaukset ovat Mock-olioita. Ilman JUnitia ei testejä voida suorittaa, joten ilman mock-olioita on mahdotonta tehdä yhtään yksikkötestiä.

**11.d** Mitä ajanhetkeä MockKelloLahde-luokan haeAjanhetki-metodi palauttaa?

- ☐ 1970-01-01 00:00:00
- ☐ 1970-01-02 11:22:33
- ☐ 2014-01-02 01:02:03

**11.e** Mikäli luot testisuiteluokan, minkä nimisellä annotaatiolla määrittelet suiteen kuuluvat testiluokat?

- ☐ @SuiteClasses
- ☐ @TestSuite
- ☐ @TestClasses

**11.f** Mihin tiedostoon Mavenia käytettäessä määritellään mm. sovellusprojektin kirjastoriippuvuudet?

- ☐ web.xml
- ☐ pom.xml
- ☐ lib.properties

**11.g** Mihin kansioon Maven-projektissa laitetaan sovelluksen toteutukseen liittyvät javakoodit?

- ☐ WebContent
- ☐ src/main/java
- ☐ src/webapp/java

**11.h** Mihin kansioon Maven-projektissa laitetaan esim. sovelluksen toteutukseen liittyvät asetustiedostot?

- ☐ src/main/resources/
- ☐ conf/
- ☐ config/

**11.i** Mihin kansioon Maven-projektissa laitetaan sovelluksen testaukseen liittyvät javakoodit?

- ☐ WebContent/test
- ☐ src/webapp/test
- ☐ src/test/java

**11.j**      **Mihin kansioon Maven-projektissa laitetaan sovelluksen webbisisältö?**

- ☐      WebContent
- ☐      src/main/webapp
- ☐      public\_html

**11.k**      **Minne maven tallentaa riippuvuuskirjastot tiedostojärjestelmässä?**

- ☐      Eclipse-workspacen sisään .metadata-kansioon
- ☐      Windows: C:\Program Files\maven\jars, Linux ja OS X: \dev\mvn\jars
- ☐      ~/.m2 (merkki "~" tarkoittaa käyttäjän kotihakemistoa)

**11.l**      **Mitä komento mvn install tekee?**

- ☐      Ei mitään, koska install-nimistä mavenin vaihetta ei ole olemassa.
- ☐      Lähettää paketoitun projektin julkiseen repositoryyn muiden sovelluskehittäjien ladattavaksi
- ☐      Asentaa käännetyt, testatut ja paketoitun projektin paikalliseen maven repositoryyn, jotta muut paikalliset projektit voivat käyttää pakettia riippuvuutenaan.

**12.a**      **Montako ilmentymää esimerkisovelluksessa 12 luodaan Auto-luokasta?**

- ☐      kaksi
- ☐      yksi
- ☐      kolme

**12.b**      **Mikä on Springin beanien default scope?**

- ☐      Admin
- ☐      Prototype
- ☐      Singleton

**12.c**      **Mitä erilaisia scopeja Spring tarjoaa beaneille?**

- ☐      user, admin, superadmin
- ☐      singleton, prototype, request, session, global session
- ☐      block, method, class, application, web-app, server, cluster



**12.d Mitä 12. esimerkisovellus tulostaa konsoliin? (ilman lokitusta)**

- ☐ mersuR1R2R3R4 (fi.agileo.spring.e12.Auto@<hashCode>)  
mersuR1R2R3R4 (fi.agileo.spring.e12.Auto@<hashCode>)
- ☐ auto  
auto
- ☐ [fi.agileo.spring.e12.Auto] (<hashCode>) 1  
[fi.agileo.spring.e12.Auto] (<hashCode>) 2

**12.e Minkälaisella xml-määrittelyllä autoilla olisi neljä rengasta, joiden numero on 1?**

- ☐

```
<bean id="rengas1" class=" fi.agileo.spring.e12.Rengas" scope="prototype" count="4"><property name="numero" value="1"/></bean>
```
- ☐

```
<bean id="auto" class=" fi.agileo.spring.e12.Auto"><property name="nimi" value="mersu"/><property name="renkaat"><list><ref bean="1"/><ref bean="1"/><ref bean="1"/><ref bean="1"/></list></property>
```
- ☐

```
<bean id="auto" class=" fi.agileo.spring.e12.Auto">
  <property name="nimi" value="mersu"/>
  <property name="jalat">
    <list>
      <ref bean="rengas1"/>
      <ref bean="rengas1"/>
      <ref bean="rengas1"/>
      <ref bean="rengas1"/>
    </list>
  </property>
</bean>
```

**13.a Minkä niminen luokka esimerkisovelluksessa 13 toteuttaa HenkiloDAO-rajapinnan?**

- ☐ HenkiloRowMapper
- ☐ HenkiloDAO
- ☐ HenkiloDAOSpringJdbcImpl

**13.b Mitä HenkiloRowMapper tekee?**

- ☐ Hakee henkilöitä tietokannasta ja muodostaa ArrayListin niistä.
- ☐ Piirtää HenkiloImg-ikoneita riveittäin GoogleMaps-kartan päälle
- ☐ Muodostaa yhdestä ResultSetin rivistä yhden Henkilo-olion.

**13.c Mikä attribuutti injektoidaan luokassa HenkiloDAOSpringJdbcImpl?**

- ☐ talleta
- ☐ haeKaikki
- ☐ jdbcTemplate

**13.d Mitä esimerkksiovellus 13 tulostaa konsoliin? (ilman lokitusta)**

- ☐ -----LISTATAAN KAIKKI-----  
MattiMikkoOutiTiinaUusi  
-----HAETAAN YKSI (ID=3)-----  
3  
Outi  
Opiskelija  
-----LISÄTÄÄN UUSI-----  
-----HAETAAN KAIKKI-----  
Matti  
Mikko  
Outi  
Tiina  
Uusi
- ☐ -----LISTATAAN KAIKKI-----  
h.getEtunimi()  
h.getEtunimi()  
-----HAETAAN YKSI (ID=3)-----  
outi.getId()  
outi.getEtunimi()  
outi.getSukunimi()  
-----LISÄTÄÄN UUSI-----  
-----HAETAAN KAIKKI-----  
h.getEtunimi()  
h.getEtunimi()  
h.getEtunimi()
- ☐   
ArrayIndexOutOfBoundsException

**13.e Mikä lokitustaso 12. esimerkksiovelluksessa on kytketty päälle?**

- ☐ fatal
- ☐ debug
- ☐ info

**13.f Mikä on tietokantayhteyksiä hallinnoivan connection poolin yhteyksien määrä sovellusta käynnistettäessä?**

- ☐ 0
- ☐ 2
- ☐ 1

**13.g Mikä on aktiivisten tietokantayhteyksien maksimimäärä?**

- ☐ 5
- ☐ 50
- ☐ 5000

**14.a** Missä tiedostossa määritellään 14. esimerkksiovelluksessa käytettävä servlet-luokka?

- ☐ spring-servlet.xml
- ☐ web.xml
- ☐ applicationContext.xml

**14.b** Mikä luokka toimii esimerkksiovelluksen 14 servlettinä ja periytyy luokasta `javax.servlet.http.HttpServlet`?

- ☐ RequestDispatcher
- ☐ org.springframework.web.servlet.DispatcherServlet
- ☐ ForwarderServlet

**14.c** Missä hakemistossa jsp-sivut sijaitsevat?

- ☐ src/main/webapp/WEB-INF/views/
- ☐ WebContent/
- ☐ WebContent/WEB-INF/

**14.d** Kuinka `welcome.jsp` on määritelty aloitussivuksi?

- ☐ `<mvc:view-controller path="/" view-name="welcome" />`
- ☐ `<property name="dataSource" ref="dataSource"/>`
- ☐ `<mvc:resources mapping="/resources/**" location="/resources/" />`

**14.e** Miten päästään käsiksi `hh_logo.jpg` tiedostoon http-pyynnön seurauksena?

- ☐ `<mvc:annotation-driven />`
- ☐ `<property name="suffix" value=".jsp" />`
- ☐ `<mvc:resources mapping="/resources/**" location="/resources/" />`

**14.f** Kuinka Springin xml-konfiguraatiossa voidaan käyttää properties-tiedoston arvoja?

- ☐ `<property name="dataSource" ref="dataSource"/>`
- ☐ `<property name="driverClassName" value="${db.driver}" />`
- ☐ `<property name="driverClassName" value="db.driver" />`

**14.g Minkälaisella annotaatiolla Spring Web MVC:ssä määritellään metodille tai luokalle url-mäppäys?**

- ☐ @RequestMapping(value="logout", method=RequestMethod.POST)
- ☐ @ValueMapping(value="uusi", method=RequestMethod.GET)
- ☐ @RequestMapping (value="/tapahtumat")

**14.h Miten saat jsp-sivulla käyttöön Springin form-tagikirjaston?**

- ☐ <%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="form" %>
- ☐ <%@ taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
- ☐ <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="form" %>

**14.i Miksi selain saa http-statuskoodin 404 (NOT FOUND), jos henkilön hakupyyntö sisältää tuntemattoman id:n, esim. "http://localhost:8080/e14/henkilot/999"? Kuinka tämä on toteutettu sovellukseen?**

- ☐ Servletti uudelleenohjaa http-pyyntönsä error.jsp-sivulle, jossa status vaihdetaan arvosta 200 arvoon 404.
- ☐ Dao heittää tilanteessa NullPointerExceptionin, johon on konfiguroitu:  
@ResponseStatus(value=HttpStatus.404)
- ☐ Dao heittää tilanteessa HenkiloaEiLoydyPoikkeuksen, johon on konfiguroitu:  
@ResponseStatus(value=HttpStatus.NOT\_FOUND)

**14.j Mistä rajapinnasta on toteutettu anonymi sisäluokka esimerkisovelluksessa 14?**

- ☐ PreparedStatementCreator
- ☐ PreparedStatement
- ☐ jdbcTemplate

**15.a Minkälaisella koodirivillä saat jsp-sivulla käyttöön springin tagikirjaston, joka mahdollistaa mm. lokalisoitujen ilmaisujen hakemisen properties-tiedostosta?**

- ☐ <%@ taglib uri="http://www.springframework.org/i18n" prefix="i18n"%>
- ☐ <%@ taglib uri="http://www.springframework.org/tags/localisation" prefix="localisation"%>
- ☐ <%@ taglib uri="http://www.springframework.org/tags" prefix="spring"%>

**15.b Minkälaisella tagilla createForm.jsp hakee sanan "Sähköposti" tiedostosta messages\_fi.properties (eli sanan "Email" tiedostosta messages\_en.properties, kun englannin kieli on valittuna)?**

- ☐ <spring:message code="henk.create.email" />
- ☐ <form:label path="sahkoposti">
- ☐ <form:errors path="sahkoposti" cssClass="Virheteksti"/>

**15.c** Minkälaisella tagilla createForm.jsp-tiedostoon on määritelty paikka, johon sukunimeen liittyvät virheviestit tulostetaan?

- ☐ <form:label path="sukunimi">
- ☐ <form:errors path="sukunimi" cssClass="Virheteksti"/>
- ☐ <spring:message code="henk.create.lastname" />

**15.d** Miten lomakkeen kentät on saatu muuttamaan taustaväriään validoinnin epäonnistuesssa?

- ☐ Määrittelemällä form-tagin style-attribuutin arvoksi "color:red".
- ☐ Määrittelemällä form:input-tagin cssErrorClass-attribuutin arvoksi "VirheellinenKentta".
- ☐ Määrittelemällä form:errors-tagin cssClass-attribuutin arvoksi "Virheteksti".

**15.e** Minkälaisella määrittelyllä tiedostossa spring-servlet.xml asetetaan resurssit messages\_fi.properties ja messages\_en.properties käyttöön.

- ☐

```
<bean id="messageSource"
class="org.springframework.context.support.ReloadableResourceBundleMessageSource">
  <property name="basename" value="classpath:messages" />
  <property name="defaultEncoding" value="UTF-8"/>
</bean>
```
- ☐ <mvc:resources mapping="/resources/\*\*" location="/resources/" />
- ☐

```
<bean id="localeResolver" class="org.springframework.web.servlet.i18n.CookieLocaleResolver">
  <property name="defaultLocale" value="fi" />
</bean>
```

**15.f** Miten sovelluksen käyttöliittymälle on asetettu oletuskieli?

- ☐

```
<bean
id="messageSource" class="org.springframework.context.support.ReloadableResourceBundleMessage
Source">
  <property name="basename" value="classpath:messages" />
  <property name="defaultEncoding" value="UTF-8"/>
</bean>
```
- ☐ <mvc:resources mapping="/resources/\*\*" location="/resources/" />
- ☐

```
<bean id="localeResolver" class="org.springframework.web.servlet.i18n.CookieLocaleResolver">
  <property name="defaultLocale" value="fi" />
</bean>
```

**15.g** Minkä nimisessä http-parametrissa selain voi lähettää sovellukselle pyynnön vaihtaa käyttöliittymän kieltä?

- ☐ kieli
- ☐ lang
- ☐ language

**15.h** Millä annotaatiolla bean määritellään validoitavaksi?

- ☐ @Validate
- ☐ @Valid
- ☐ @Validoi

**15.i** Millä annotaatiolla voidaan tarkistaa merkkijonon pituus?

- ☐ @NumberOfChars
- ☐ @Size
- ☐ @Pituus

**15.j** Minkälaisella annotaatiolla voit validoida muuttujan säännöllisen lausekkeen (regular expression) perusteella?

- ☐ @Pattern
- ☐ @RegularExpression
- ☐ @RegExp

**15.k** Missä tiedostossa määritellään, että annotaatio @Paakaupunkiseutu on käytettävissä?

- ☐ HenkiloBean.java
- ☐ PaakaupunkiseutuValidator.java
- ☐ Paakaupunkiseutu.java

**15.l** Missä tiedostossa hoidetaan @Paakaupunkiseutu-annotaatiolla merkittyjen kenttien validointi?

- ☐ HenkiloBean.java
- ☐ PaakaupunkiseutuValidator.java
- ☐ Paakaupunkiseutu.java

**16.a** Mihin osoitteeseen sisäänkirjautumislomakkeen tiedot lähetetään?

- ☐ j\_spring\_security\_check
- ☐ login\_check
- ☐ kirjaudu

**16.b Mitä url:a selaimen tulee kutsua, jotta käyttäjä kirjataan ulos sovelluksesta?**

- ☐ logout
- ☐ j\_spring\_security\_logout
- ☐ spring\_security\_logout

**16.c Kuinka sisäänkirjautuneen käyttäjän käyttäjätunnus saadaan sivulle näkyviin?**

- ☐ `<sec:authentication property="principal.username"/>`
- ☐ `<sec:authorize access="hasRole('ROLE_ADMIN')">`
- ☐ `<sec:authentication print="username"/>`

**16.d Miten main-sivulla "admin tools"-linkki tulostetaan ainoastaan järjestelmän pääkäyttäjille?**

- ☐ `<c:if test="role is admin"><p><a href="admin/tools">Admin tools</a></p></c:if>`
- ☐ `<sec:authentication login="ROLE_ADMIN"><p><a href="admin/tools">Admin tools</a></p></sec:authentication>`
- ☐ `<sec:authorize access="hasRole('ROLE_ADMIN')">  
<p><a href="admin/tools">Admin tools</a></p></sec:authorize>`

**16.e Minkälaisella xml-koodilla voit määritellä käyttäjät ja niiden roolit suoraan springin xml-konfiguraatiotiedostoon?**

- ☐

```
<sec:authentication-manager>
  <sec:authentication-provider>
    <sec:users="user:userpassu; admin:adminpassu" />
  </sec:authentication-provider>
</sec:authentication-manager>
```
- ☐

```
<sec:authentication-manager>
  <sec:authentication-provider>
    <sec:user-service>
      <sec:user name="user" password="userpassu" authorities="ROLE_USER" />
      <sec:user name="admin" password="adminpassu" authorities="ROLE_ADMIN" />
    </sec:user-service>
  </sec:authentication-provider>
</sec:authentication-manager>
```
- ☐ `<sec:global-method-security pre-post-annotations="enabled" />`

**16.f Mikä Spring Security Frameworkin luokka on vastuussa salasanan kryptaamisesta, suolan generoimisesta ja niiden liittämisestä peräkkäin?**

- ☐ Salaja
- ☐ StandardPasswordEncoder
- ☐ PasswordEncoder

**16.g Miten sovellukseen on määritelty tietokantapohjainen autentikointi?**

- ☐ `<sec:authentication-manager><sec:authentication-provider><sec:password-encoder ref="encoder" /><sec:jdbc-user-service data-source-ref="dataSource"users-by-username-query="select username, password_encrypted, enabled from webuser2 where username=? and enabled = 1"authorities-by-username-query="select u.username, a.role from webuser2 u JOIN webuser2_authority ua ON (u.id = ua.webuser2_id) JOIN authority a ON (a.id = ua.authority_id) WHERE u.username = ?" /></sec:authentication-provider></sec:authentication-manager>`
- ☐ `db.driver=org.mariadb.jdbc.Driver  
db.url=jdbc:mysql://localhost/projekti  
db.username=projekti  
db.password=toFEmC72t`
- ☐ `<sec:http auto-config="true" use-expressions="true"> ... </sec:http>`

**16.h Mihin käyttäjäkohtainen salasanan kryptaamisessa käytetty suola on tallennettu tietokannassa?**

- ☐ Suola on omassa kentässä tietokannassa
- ☐ Se on liitettyinä kryptattuun salasanaan
- ☐ Suola on kaikilla käyttäjillä sama

**16.i Mikä estää käyttäjätunnusta matti pääsemästä sivulle secure/admin/tools?**

- ☐ `authentication-failure-url="/loginfail"`
- ☐ `<sec:intercept-url pattern="/secure/*" access="isAuthenticated()" />`
- ☐ `<sec:intercept-url pattern="/secure/admin/**" access="hasRole('ROLE_ADMIN')"/>`

**16.j Minkälaisella xml-määrittelyksellä Spring Security Framework saadaan lukemaan myös @PreAuthorize-annotaatiot?**

- ☐ `<sec:authentication-manager>`
- ☐ `<sec:http auto-config="true" use-expressions="true">`
- ☐ `<sec:global-method-security pre-post-annotations="enabled" />`

**16.k Miksi vain sisäänkirjautuneet käyttäjät pääsevät MiscServiceImpl-luokasta löytyvään metodiin palveluSisaankirjautuneille()?**

- ☐ `user.isAuthenticated()`
- ☐ `@PreAuthorize("isAuthenticated()")`
- ☐ `@IsAuthenticated`



**17.a Mikä kirjasto hoitaa Java-olioiden kääntämisen json-muotoon?**

- ☐ spring-context
- ☐ jackson-databind
- ☐ json-converter

**17.b Millä annotaatiolla merkitään, että controllerin metodin palautusarvo kirjoitetaan suoraan http responsen body-osioon.**

- ☐ @ResponseBody
- ☐ @BodyOfResponse
- ☐ @Response

**17.c Mitä JavaScript-kirjastoa on käytetty apuna 17. esimerkkisovelluksen front-endissä?**

- ☐ AngularJs
- ☐ JQuery
- ☐ BaconJS

**18.a Minkä niminen projekti määritellään tyypillisesti Spring Boot -projektin parentiksi Mavenin konfiguraatioissa?**

- ☐ spring-boot-starter-parent
- ☐ spring-boot-starter-web
- ☐ spring-boot-maven-plugin

**18.b Millä Maven-riippuvuudella saat Spring Boot -projektista webbiprojektin=**

- ☐ spring-boot-starter-parent
- ☐ spring-boot-starter-web
- ☐ spring-boot-maven-plugin

**18.c Mikä Maven-plugin auttaa ajettavan jar-tiedoston muodostamisessa Spring Boot -projektissa?**

- ☐ spring-boot-starter-parent
- ☐ spring-boot-starter-web
- ☐ spring-boot-maven-plugin

**18.d Minkälainen metodikutsu main-metodissa käynnistää Spring Boot -sovelluksen?**

- ☐ SpringApplication.run()
- ☐ Thread.run();
- ☐ Runnable.run();

**19.a Millä annotaatiolla merkitään Springin Java-konfiguraatioluokka?**

- ☐ @JavaConf
- ☐ @ConfigurationClass
- ☐ @Configuration

**19.b Mistä luokasta Springin Java-konfiguraatioluokka tulee periyttää Spring Web MVC-sovelluksessa?**

- ☐ WebMvcConfiguration
- ☐ WebMvcConfigurerAdapter
- ☐ WebConfigurations

**20.a Millä RequestMappingillä Controller konfiguroidaan portletin view-modeen?**

- ☐ @RequestMapping
- ☐ VIEW
- ☐ spring-webmvc-portlet

**20.b Millä annotaatiolla kontrollerin metodi määritellään palvelemaan portletin Render-vaihetta?**

- ☐ @RequestMapping
- ☐ @Renderer
- ☐ @RenderPhase

**20.c Mikä Maven-riippuvuus vaaditaan Spring Portlet MVC -sovelluksen tekemiseen?**

- ☐ spring-context
- ☐ spring-tx
- ☐ spring-webmvc-portlet