



שאלות לדוגמה:

1. למה להשתמש ב *hibernate*?
2. למה משמשת המחלקה *Session*?
3. מהם היתרונות של השימוש ב *HQL*?
4. הסבר בקצרה מה זה *Reflection* ומה השימוש במנגנון הזה.
5. הסבר מה השימוש ב *XSL*?
6. מה השימוש של *XPath*?
7. מהם ההבדלים בין *SAX* ל-*DOM*?
8. אילו *XML Parsers* אנו מכירים?
9. מהם *scriptlets*?
10. מהו *JSP*, מהו *SERVLET* ומהם ההבדלים בין השניים?
11. מה הפלט עבור קטעי הקוד הבאים? (*JSP*)

```
<HTML>
<BODY>
  <%
    String test = "tomorrow.";
  %>
  Hello! The test takes place
  <%
    System.out.println(" "+test);
  %>
  ;)
  <br>
  Good luck to everyone!
</BODY>
</HTML>
```

.12

```
<HTML>
<BODY>
  <%
    String test = "tomorrow.";
  %>
  Hello! The test takes place
  <%
    out.println(" "+test);
  %>

  <br>
  Good luck to everyone!
</BODY>
</HTML>
```

הצעת פתרון:

- למה להשתמש ב hibernate?

בתוכניות מורכבות המחזיקות מידע רב, נרצה להשתמש במסד נתונים ולא להתעסק ברמת מבנה הנתונים. נרצה אף להפריד בין שכבת מסד הנתונים לשכבת האפליקציה שעושה בו שימוש. נרצה שהקוד שלנו לא ישתנה אם החלטנו להחליף למסד נתונים חדש בעל שפת שאלות אחרת. בעיה יותר חמורה היא שאנו נרצה לעבוד עם **אובייקטים** כחלק מתוכנית מונחית עצמים – ולא לעבוד עם טבלאות.

מתאם אוטומטי בין סכמה רלציונית למחלקה ובין תוכן הטבלה למופע של מחלקה זו.

- למה משמשת המחלקה Session?

כדי שנוכל לשייך מידע עבור כל משתמש. זהו אובייקט מידע הנוצר עבור כל משתמש, מבנה נתונים לכל דבר הניתן לאחסן בו ולחלץ ממנו מידע, לאחר timeout מסוים המידע נמחק.

- מהם היתרונות של השימוש ב HQL?

שפת שאלות שדומה בכוונה ל SQL. יכולת הבנה של ירושה, פולימורפיזם, קשר אסוציאטיבי.

- הסבר בקצרה מה זה Reflection ומה השימוש במנגנון הזה.

Reflection הינו מנגנון לקבלת מידע על תוכן המחלקה: תכונות, שיטות, הרשאות וכד', מבלי שתהייה לנו גישה לקוד עצמו. מנגנון גנרי של שיקוף המידע. מנגנון זה מאפשר לנו להפעיל שיטות באופן דינאמי מבלי לדעת מראש מה יש בתוך המחלקה. (זיהוי השיטה ע"י מנגנון ה- reflection והפעלתה באמצעות invoke).

- הסבר מה השימוש ב XSL?

אם אי פעם נרצה להציג את המידע שלנו בפורמט HTML נוכל להשתמש בטרנספורמציה הזאת כדי להפוך תגיות של XML לתגיות של HTML.

- מה השימוש של XPath?

יאפשר לנו לנווט בתוך מסמך ה-xml ואף לעשות עליו חיפושים ושאלות.

- מהם ההבדלים בין SAX ל-DOM?

DOM שומר את כל אלמנטים ה XML בזיכרון, בעוד ש SAX מפרסר אלמנט-אלמנט. מה שאומר ש DOM צורך יותר זיכרון אבל יש לו יותר יכולות (למשל פרסור מהסוף להתחלה) מאשר SAX.
SAX - קורא את המסמך סדרתית.
DOM קורא את כל המסמך, ומכין עץ של המסמך בזיכרון. אם יש מגבלת זיכרון עדיף להשתמש ב - SAX.
אם חשוב לך לעבור על המסמך הרבה פעמים קדימה/אחורה ולעשות מניפולציות עליו, עדיף DOM.

- אילו XML Parsers אנו מכירים?
DOM, JDOM, SAX, JAXP

- מהם scriptlets?
בלוקים שלמים של קוד בג'אווה המושתלים בקבצי JSP.

- מהו JSP, מהו SERVLET ומהם ההבדלים בין השניים?
JSP – הינו קוד HTML שניתן לשלב בו קוד ג'אווה (בעזרת script lets לדוג'). בזמן runtime.
SERVLET – קוד ג'אווה שבתוכו משולב HTML.

- הפלט הינו:

Hello! The test takes place ;)
Good luck to everyone!

- הפלט הינו:

Hello! The test takes place tomorrow.
Good luck to everyone!

