

Programozási technológiák vizsgakérdések (2018)

1. **Java:** Fogalmak: Java virtuális gép, platform, Java platform, Java SE, JRE, JDK.
2. **Java:** A Java SE kiadási modelljének változása.
3. **Java 9:** *Projekt Jigsaw*: elsődleges célok, megbízható konfigurálás és erős egységbezárás. A moduláris JDK.
4. **Java 9:** Modul fogalma. Modul deklarációk, a `requires` és `exports` direktíva.
5. **Java 9:** Olvashatóság. Névtelen modulok.
6. **Java 9:** Modul útvonalak. JMOD állományok. Szerkesztés. Moduláris futásidejű lemezképek.
7. **Annotációk (1):** Annotáció fogalma. Annotációk lehetséges felhasználásai. A `java.lang` csomag annotációi: 3 szabadon választott annotáció típus ismertetése.
8. **Annotációk (2):** Annotációk szerkezete. Annotációk fajtái (közönséges, egyelemű, jelölő). Hol alkalmazható annotáció?
9. **Annotációk (3):** Meta-annotációk: a `java.lang.annotation` csomag annotációi típusai. Annotáció típus deklarálása.
10. **JDBC (1):** JDBC meghajtó és fajtái. A `java.sql.Driver` interfész és a `java.sql.DriverManager` osztály, JDBC meghajtók betöltése.
11. **JDBC (2):** Adatbázis kapcsolat létrehozása: a `java.sql.DriverManager` osztály és a `javax.sql.DataSource` interfész. Kapcsolatok gyorsítótárazása (*connection pooling*).
12. **JPA (1):** Fogalmak: perzisztencia, adatelérési objektum (DAO), szakterületi modell, vérszegény szakterületi modell, POJO, JavaBean, objektum-relációs leképezés, paradigma ütközés (*object-relational impedance mismatch*).
13. **JPA (2):** Entitás fogalma. Entitás osztályok. A `javax.persistence.EntityManager` interfész. Perzisztencia kontextus. Perzisztencia egység. A `javax.persistence.EntityManagerFactory` interfész.
14. **JPA (3):** `EntityManager` műveletek: `persist`, `find`, `remove`, `detach`, `merge`, `clear`. Perzisztencia műveletek kaszkádolása. Leválasztott entitások.
15. **JPA (4):** Elsődleges kulcsok és entitás identitás. Elsődleges kulcs generálása, generálási stratégiák.
16. **JPA (5):** Objektum-relációs leképezés megvalósítása a gyakorlatban: a Java kóddal és/vagy UML osztálydiagrammal adott osztályok leképezése SQL táblákra.
17. **UML (1):** Fogalmak: modell, metamodell, szakterület specifikus nyelv. *Meta Object Facility* (MOF): négyrétegű metamodell architektúra.
18. **UML (2):** Diagram elemek: csomag, kulcsszó, sztereotípa, megjegyzés, függőség, megszorítás.
19. **UML (3):** Osztálydiagramok fajtái. Osztályok jelölésmódja, attribútumok és műveletek. Statikus attribútumok és műveletek. Absztrakt osztály.
20. **UML (4):** Asszociációk: jelentés, jelölésmód, asszociáció vég, navigálhatóság.
21. **UML (5):** Egész-rész kapcsolat: aggregáció és kompozíció. Általánosítás. Interfész, interfész realizáció.
22. **Minták a szoftverfejlesztésben (1):** Fogalmak: minta és részei, mintakatalógus, tervezési minta, programozási idióma.
23. **Minták a szoftverfejlesztésben (2):** Architektúrális minta fogalma, a modell-nézet vezérlő (MVC) architektúrális minta bemutatása.
24. **Minták a szoftverfejlesztésben (3):** Antiminta fogalma, a massa és a spagetti kód szoftverfejlesztési antiminták bemutatása.
25. **Tervezési minták:** Az adott létrehozási/szerkezeti/viselkedési minta céljának megfogalmazása és a minta gyakorlati használatának bemutatása adott Java programkód és UML osztálydiagram alapján. Innen származó programkódok lesznek kiadva: <https://github.com/iluwatar/java-design-patterns>

A következő tervezési minták ismerete szükséges:

- Létrehozási minták: elvont gyár, építő, gyártó metódus, objektumkészlet, prototípus, egyke
- Szerkezeti minták: illesztő, összetétel, díszítő, homlokzat, helyettes
- Viselkedési minták: felelősséglánc, bejáró, megfigyelő, stratégia, sablonfüggvény, látogató

26. **OO alapelvek (1):** A DRY elv, ismétlések fajtái. A KISS elv. Demeter törvénye.
27. **OO alapelvek (2):** Vonatkozások szétválasztása. Vonatkozás fogalma, vonatkozások fajtái.
28. **OO alapelvek (3):** GoF alapelvek. Az öröklődés (fehér dobozos újrafelhasználás) és az objektum-összetétel (fekete dobozos újrafelhasználás) előnyei és hátrányai.
29. **OO alapelvek (4):** SOLID alapelvek: az egyszeres felelősség elve, a elv megsértésének magyarázata adott Java programkód és/vagy UML osztálydiagram alapján, a példa az elvnek megfelelő javítása.
30. **OO alapelvek (5):** SOLID alapelvek: a nyitva zárt elv, a elv megsértésének magyarázata adott Java programkód és/vagy UML osztálydiagram alapján, a példa az elvnek megfelelő javítása.
31. **OO alapelvek (6):** SOLID alapelvek: a Liskov-féle helyettesítési elv, az elv megsértésének magyarázata adott Java programkód és/vagy UML osztálydiagram alapján, a példa az elvnek megfelelő javítása.
32. **OO alapelvek (7):** SOLID alapelvek: az interfész szétválasztási elv, vastag interfész, interfész szennyezés, az interfész szétválasztási elv megsértésének magyarázata adott Java programkód és/vagy UML osztálydiagram alapján, a példa az elvnek megfelelő javítása.
33. **OO alapelvek (8):** SOLID alapelvek: a függőség megfordítási elv, az elv megsértésének magyarázata adott Java programkód és/vagy UML osztálydiagram alapján, a példa az elvnek megfelelő javítása.
34. **Függőség befecskendezés:** Definíció. Fogalmak: függőség, függő, objektum gráf, befecskendezés, DI konténer. Megoldások függőség befecskendezés megvalósítására.
35. **Szoftverlicencek (1):** Fogalmak: nem szabad szoftver, szabad szoftver, nyílt forrású szoftver, szabad és nyílt forrású szoftver, kereskedelmi szoftver, *freeware*, *shareware*, közkincs.
36. **Szoftverlicencek (2):** Fogalmak: szoftverlicenc, megengedő licenc, *copyleft*, *copyleft* licenc, erős és gyenge *copyleft* licenc, licenc kompatibilitás.
37. **Szoftverlicencek (3):** GNU GPL, GNU GPL-es szoftverek használata. GNU AGPL.
38. **Szoftverlicencek (4):** GNU LGPL, a GNU LGPL problémái.
39. **Szoftverlicencek (5):** Miért fejlesszünk üzletszerűen szabad és nyílt forrású szoftvert? Nyílt forrású üzleti modellek: kettős licencelés, többszörös licencelés.
40. **Szoftverlicencek (6):** Licencek egyéb alkotásokhoz. *Creative Commons*-licencek.
41. **Tiszta kód (1):** Kódújrászervezés fogalma. Nevek használatára vonatkozó szabályok.
42. **Tiszta kód (2):** Függvényekre vonatkozó szabályok.
43. **Tiszta kód (3):** Jó és rossz megjegyzések fajtái.
44. **Tiszta kód (4):** Forráskód formázás. Függőleges formázás: az újság metafora. Vízszintes formázás. Kódolási konvenciók.
45. **Tiszta kód (5):** Tiszta kód a gyakorlatban: adott Java programkód hibáinak megállapítása és a hibák javítása.