# Alkalmazás fejlesztő specializáció

**záróvizsga kérdések**

**BSc, 2020.nov. 1-től érvényben**

Az alábbi tételek egy beszélgetés formájában kerülnek kikérdezésre egy gyakorlati példán keresztül.

1. Relációs adatbázis tervezés módszerei.
* Melyek a relációs adatmodellek fő jellemzői, komponensei? Milyen szabályok alkalmazásával képezhetünk le szemantikai modellt relációs modellbe?
* Definiálja a funkcionális függőséget! Ismertesse az Armstrong axiómákat! Kulcs, szuperkulcs fogalma. Milyen normálformákat ismer? Normalizálás lépései.
* primary key, foreign key, unique key, index fogalma
* index kezelés, bfa felépítése
1. Ismertesse a relációs adatbáziskezelés módszereit tranzakciókezeléssel és a SQL utasításokkal!
* Mi a tranzakció, mi a jelentősége? Mik az ACID elvek? Milyen utasításokkal jelezhetjük a tranzakció kezdetét és végét? Mi a konkurens működés lényege, milyen zárakat (lock) alkalmazhatunk a konkurens végrehajtásból fakadó problémák kiküszöbölésére? Milyen izolációs szinteket definiálhatunk?
* Mutassa be az SQL nyelvet! Ismertesse a DDL és DML utasításokat!. A felsoroltaknak mi a lényege, milyen műveletet végzünk velük? Ismertesse a DDL és DML közti különbséget tranzakciókezelés szempontjából.
* Milyen adattípusok használatát támogatják az adatbáziskezelő rendszerek és az SQL nyelv? SQL programozás: tárolt eljárások (Stored Procedures), felhasználói függvények (User Defined Functions), triggerek jelentősége, használata.
1. Ismertesse a verziókezelők fajtáit, nevezzen meg verziókezelő programokat!
* Fejtse ki a különbséget az elosztott és a központi verziókezelés között.
* Gyakorlati példákkal mutassa be az alábbi fogalmak lényegét: branch, tag, commit, merge, rebase, fetch. Mutassa be a Git Flow folyamatát.
* Mutassa be a tesztelés lehetőségeit és life-cycle folyamatok kezelését a forrás-kódra vonatkozóan.
1. Ismertesse a réteges alkalmazás fejlesztés lényegét, sorolja fel annak előnyét, milyen rétegeket tud megkülönböztetni, melyiknek mi a feladata?
* Prezentálja egy példa alkalmazás felépítését
1. Ismertesse a webszolgáltatások típusait, melyiknek mi a formátuma, lényege, eszköze.
* Milyen publikus web szolgáltatásokat ismer?
* Mutassa be a SOA és annak elveinek lényegét
* Mi a különbség a REST és az RPC között?
1. Ismertesse az objektum orientált programozás fogalmát, alapelveit!
* Ismertesse az objektum orientáltság kapcsán használt operátorokat
* SOLID elvek
* Az osztályok, objektumok között milyen kapcsolatok lehetségesek? Részletezze!
* Hasonlítsa össze Java, C# vagy JavaScript nyelveket.
1. Ismertesse a szerver oldali alkalmazások működését és felépítését! (Választhat az Enterprise Java, Spring, NodeJS vagy ASP.NET technológiák közül.)
* Ismertesse a szálkezelést, a kritikus pontjait és azok megoldási lehetőségét, Reflection és Annotation lényegét!
* Ismertesse a fentiek felhasználási területeit.
* Milyen keretrendszerek milyen módon épülnek a fenti apparátusokra.
1. Ismertesse a nagy szoftverrendszerek fejlesztési modelljeit! (Vízesés, evolúciós, iteratív, spirál modell, RUP)
	* Mi jellemzi az agilis módszertanokat? Milyen alapelveket vallanak az agilis módszerek? Mi a Scrum lényege?
2. Ismertesse a keretrendszerek használatához kapcsolódó tervezési és implementációs fogalmakat.
* Ismertesse a Dependency Injection programtervezési mintát, milyen scope-okat ismer?
* Mi az az IOC container, mi a működési elve?
* Mutassa be a Spring keretrendszert! Milyen modulokat ismer?
1. Ismertesse a jogosultságkezelési biztonsági elveket rétegenként szoftverfejlesztés kapcsán!
* Ismertesse a SQL injection lényegét, milyen módszereket ismer ennek elkerülésére?
* Ismertesse a hálózati adatbiztonság kihívásait és megoldásukat az alkalmazás fejlesztés viszonylatában. Mutassa be az AES, RSA, DH algoritmusok felhasználását, példákon keresztül, pl. webes szolgáltatáshoz tanúsítvány kérés, jelszó titkosítás, stb.
* Mutassa be egy választott szerver oldali rendszerrel a jogosultság kezelést.