

UML analízis osztályok

1. Definiáld a (RUP által bevezetett) analízis osztályokat!
2. Melyek az egyes analízis osztály fajták főbb jellegzetességei (attribútumok, relációk, metódusok, függőségek, stb.)?
3. Hogy kapcsolódnak egymáshoz az analízis osztályok és mit jelentenek ezek a kapcsolódások architektúrális szempontból?
4. Miben különbözhet jellegzetesen egymástól egy analízis szintű (PIM) és egy tervezési (design, PSM) szintű osztálydiagram?

UML usecase modellezés

1. Mik azok a használati esetek (use case, UC)? Hogyan épül fel egy használati eset leírása?
2. Milyen kapcsolatban lehetnek az egyes aktorok, UC-ek?
3. Hogyan fejthető ki részletesebben egy UC?
4. Hogyan követhető nyomon az egyes UC-ek (avagy mi a hatásuk) a tervezés későbbi fázisaiban?

UML Object Constraint Language

1. Mi az OCL célja? Jellegzetesen hol (milyen diagramokban) használunk OCL kényszereket UML modellekben?
2. OCL-ben hogyan megy a navigáció és a halmazelméleti műveletek végrehajtása?
3. Hogyan értékelhetők ki futási időben az egyes OCL kényszerek egy JBoss Seam-alapú implementáció felett (hogyan implementáljuk, mikor ellenőrizzük, melyek a nehezen ellenőrizhető - problematikus - kényszerek)?
4. Mik az OCL alkalmazásának korlátai?

UML szekvencia diagramok, állapottérképek, architektúrális modellezés

1. A tervezés milyen fázisaiban készítünk szekvencia diagramokat, és mely tervezési fázisban mire használhatjuk azokat?
2. UML szekvencia diagramok segítségével mutasd meg, hogy hol végezhetünk (adat)validációt egy JBoss Seam-alapú architektúrában? Melyek az egyes validációs pontok alkalmazásának előnyei, hátrányai? Hogyan használhatók fel az egyes UML diagramok egy JBoss Seam-alapú implementáció során adatvalidációra?
3. Architektúratervezés az UML összetett struktúra (Composite Structure) diagramjai segítségével (komponensek, interfészek, portok, kapcsolatok fogalma).
4. Mik azok a protokoll statechartok és miben különböznek a hagyományos állapottérképektől? Milyen módon valósítható meg az elosztott komponensek közötti kommunikáció?

Domain-specifikus modellezési nyelvek

1. Mi szükséges egy modellezési nyelv definiálásához? Mit értünk absztrakt szintaxis alatt? Mi a konkrét szintaxis? Miért van rá szükség?
2. Mit értünk jólformáltsági kényszer alatt? Mik a legtipikusabb (sok nyelvben előforduló) kényszerek? Hogyan lehet egy nem tipikus (nyelvspecifikus) kényszert kifejezni (példával)?
3. Értelmezhető-e példányosítási ill. finomítási viszony éltípusok (asszociációk, relációk) között? Mi következik ebből a végpontokra nézve?
4. Mik az absztrakt szintaxis és konkrét jelölésrendszer közötti viszony multiplicitása? Miért?
5. Mik az absztrakt modell és egy rögzített jelölésrendszer szerinti konkrét megjelenése közötti viszony multiplicitása? Miért?
6. Mutassa be (szabadon választva) a MOF vagy az EMF modellezés metaszintjeit! Illusztrálja egy egyszerű Jedi-adatbázis (név, mester-padawan viszony) példán keresztül!
7. Mit nevezünk többszintű (multi-level) metamodellezésnek? Mire való, mi az előnye? Milyen formalizmust ismer, ami képes erre?
8. Milyen kétféle módon adható meg egy viselkedésmodellezési nyelv dinamikus szemantikája? Melyik hogy működik?

Eclipse Modeling Framework

1. Ismertesse a legalapvetőbb Ecore struktúrákat, azaz az Ecore metamodelljét! Írjon egy egyszerű metamodellt egy személygépjárműhöz.
2. Ismertesse az EMF fejlesztési workflowt, kitérve, hogy egyes pontokban milyen elemek jönnek létre (file-ok, modellek, forrás kód, stb.)
3. Milyen különböző módokon lehet ecore modellt létrehozni?
4. Ismertesse a genmodelt, mire használják és milyen jellegű attribútumokat lehet vele beállítani?
5. Milyen projektek generálhatóak genmodelből és ezek mire szolgálnak?
6. Ismertesse a generált EMF-model-t. Térjen ki a generált Java kódban alkalmazott modelltárolási megoldásokra. Egy egyszerű példán keresztül ismertesse a generált Java osztályok és Interfészek kapcsolatát.
7. Mire szolgálnak a különböző generált EFactory-k?
8. Milyen módon lehet reflektíven modelleket módosítani és lekérdezni?
9. EMF esetén mi tekinthető absztrakt szintaxisnak és mi konkrétnek?

Kódgenerálás

1. Ismertesse a 3 legelterjedtebb kódgenerálási megközelítést. Mik az erősségeik és gyengeségeik?
2. Mi az AST? Miért fontos a kódgenerálás körében és milyen előnyöket/hátrányokat hoz be a használata.
3. Mik a generátor modell használatának előnyei és hátrányai?
4. Ismertessen legalább 3, a kódgenerálás körében ismert problémát és adjon rájuk megoldási ötleteket.
5. Milyen szabályok alapján származtathatók egy PSM szintű osztálydiagramból JBoss Seam interfészek és implementációs osztályok? Milyen funkcionalitás támogatásáig lehet eljutni kódgenerálásban (pl. EMF)?
6. Definiálja a refactoring és a reverse engineering fogalmát! A HF fejlesztése során milyen refactoring lépésekkel találkozhatott?

Fejlesztési módszertanok (RUP, XP, SPEM)

1. Mondjon példákat a szoftverprojektekben előforduló leggyakoribb problémákra, szimptomákra!
2. Mondjon példákat a szoftverprojektekben előforduló leggyakoribb problémák, szimptomák okaira!
3. Mik az iteratív fejlesztési módszertanok előnyei a szekvenciális fejlesztéssel szemben? Mire kell figyelni a projekttervezés során?
4. Mik a komponens alapú architektúra legfontosabb előnyei?
5. Soroljon fel néhány, a szoftverfejlesztésben használatos metrikát!
6. Mire való a változáskezelés (Change Control, Change Management)?
7. Mi a RUP?
8. Mit jelent a RUP esetén a "use case vezérelt" tulajdonság?
9. Mik az MDA (Model-driven Architecture) előnyei projektmenedzsment szempontból?
10. Mik egy tipikus XP projekt jellemzői?
11. Sorolja fel az XP 12 szabályát!
12. Mik az XP legfőbb előnyei és hátrányai?
13. Mit jelent az "agilitás" a szoftverprojekt-menedzsmentben? Mondjon példákat agilis fejlesztési módszertanokra!
14. Hasonlítsa össze a vízésés, RUP és XP módszertanokat!
15. Mik azok a folyamat minták? Mik a legfontosabb, jellemző tulajdonságaik?
16. Mi a SPEM? Melyik metaszinten helyezkedik el? Mire használható?

Modelltranszformációk, gráftranszformációk

1. Ismertesse azon területeket, ahol a modelltranszformáció jól hasznosítható! Adjon példát egy konkrét transzformációs nyelvre!
2. Rajzolja fel és ismertesse egy általános modelltranszformációs keretrendszer felépítését!
3. Sorolja fel az ismert modelltranszformációs megközelítéseket! Miben különböznek és miben hasonlítanak?
4. Ismertesse a gráftranszformáció alapjait! Írjon fel egy egyszerű szabályt és mutassa be, hogyan alkalmazható egy modellen!
5. Ismertesse a gráftranszformációs formalizmus kiterjesztésének lehetőségeit! Ezek közül adjon példát legalább egyre.
6. Ismertesse a gráftranszformációs rendszereket (GTS)! Milyen rendszereket lehet ezzel a módszerrel leírni?
7. Ismertesse a három legelterjedtebb gráfmintaillesztési (graph pattern matching) stratégiát! Miben különböznek és miben hasonlítanak?
8. Ismertesse az EMF-IncQuery rendszer architekturális felépítését! Milyen szerepet játszik a VIATRA a lekérdezések fejlesztése során?
9. Hogyan használható az EMF-IncQuery modellalapú fejlesztőeszközökben? Milyen gyakori funkciók támogatására használható? Mondjon pár példát hasonló technológiákra! Miben hasonlítanak és különböznek az EMF-IncQuery keretrendszerrel?