

Kérdés: Mely állítások igazak a CRC kártyákra?
Osztályok meghatározása nem diagramok alapján, hanem táblázatos lapok segítségével.
UML-ben hibajavítás során használhatóak
A leírásban csak metódusokat kell szerepeltetni.
A leírásban nem metódusokat és attribútumok, hanem az osztályokhoz rendelhető felelősségek vannak.

Kérdés: Osztálydiagramok készítésének milyen fázisai vannak?
Elemzési osztálydiagram
Kollaborációs osztálydiagram
Tervezési osztálydiagram
Osztálydiagram a tervezési minták kialakításához
Megvalósítási osztálydiagram
Futásidejű osztálydiagram

Kérdés: UML osztálydiagramon hogyan kell értelmezni két osztály közötti vonalat?
kooperáció
asszociáció
aggregáció
kompozíció

Kérdés: UML osztálydiagram számosság jelölésére melyek a szintaktikailag helyes válaszok?
0..1
*..0
*
0..*
5
0..3
Egyik sem, a számosság másképpen van jelölve.

Kérdés: Osztálydiagram esetén mire jó a fogalmi szótár?
Magyarázatot ad a modellben felhasznált metódusokra, tevékenységdiagramokkal.
Magyarázatot ad a modellben felhasznált fogalmakra, szinonímákkal, fogalmi környezettel.

Kérdés: Jelölje az igaz állítást!
Osztálydiagramon az asszociáció kapcsolatot jelöl, a kompozíció tulajdonlást.
Osztálydiagramon a kompozíció kapcsolatot jelöl, az asszociáció tulajdonlást.

Kérdés: UML osztálydiagramon mi a kompozíció jele?
Fekete rombusz végű vonal, rombusz a tartalmazó oldalon
Fekete rombusz végű vonal, rombusz a tartalmazott oldalon
Nyílt fejű nyíl
Telt fejű nyíl, az ősz felőli oldalon

Kérdés: Melyik (osztálydiagram) kapcsolattípusra igaz az alábbi állítás:
A tartalmazó objektum megszűntével az összes részobjektum is felszámolódik
Öröklődés
Kompozíció
Asszociáció
Kooperáció

Kérdés: UML osztálydiagram esetén mi az osztály rajzjele?
Lekerekített sarkú téglalap
3 részre osztott téglalap
Ellipszis

Kérdés: Mely állítások igazak osztálydiagram osztály rajzelemre?
3 részre osztott téglalap, benne osztálynév, aktor és use case lista
Egyszerű téglalap, benne use case elnevezése

Egyszerű téglalap, benne az osztály neve
3 részre osztott téglalap, benne osztálynév, attribútumok és műveletek

Kérdés: Osztálydiagramon mit jelent a dőlt betűs osztálynév?
Statikus osztály, azaz nem lehetnek példányai
Absztrakt osztály, azaz nem lehetnek példányai
Controller típusú osztály

Kérdés: Osztálydiagramon mit jelent az <<abstract>> sztereotípia?
Olyan osztályt jelöl, melyet nem lehet további öröklési hierarchiában felhasználni.
Adatbázis interfészt jelöl
Olyan osztályt jelöl, melynek nem lehetnek példányai
Olyan osztályt jelöl, melynek szigorúan csak egy példánya létezhet

Kérdés: Osztálydiagramon mit jelent az <<entity>> sztereotípia?
A felhasználó számára fontos entitások jelölésére szolgál
Azon osztályok jelölése, melyek attribútumait perzisztálni kell
Az adatbázisba írást megvalósító osztályok jelölése

Kérdés: Osztálydiagramon mit jelent az <<boundary>> sztereotípia?
A felhasználó számára fontos entitások jelölésére szolgál
A felhasználói felülethez tartozó osztályok jelölése ez
Olyan osztályt jelöl, melyet nem lehet további öröklési hierarchiában felhasználni.
Statikus osztály, azaz nem lehetnek példányai

Kérdés: Osztálydiagramon mit jelent az <<uses>> sztereotípia?
Öröklődést
Interfész használatot
Aktor általi használatot

Kérdés: Osztálydiagramon mit jelent az <<implements>> sztereotípia?
Use case megvalósítást
Interfész megvalósítást
Absztrakt osztály megvalósítást

Kérdés: Osztálydiagramon mit értünk interfész alatt?
Olyan szerződést, mely metódus szignatúrákat definiál
Absztrakt osztályt
Olyan csatlakozási pontot, mely segítségével külső rendszerekből kérhetünk le adatokat

Kérdés: Osztálydiagramon hogyan jelöljük az interfészt?
Szaggatott ellipszissel
<<implements>> sztereotípiával
<<interface>> sztereotípiával
Körrel illetve gömbcsuklóval

Kérdés: Osztálydiagramon mi a különbség egy interfész és egy absztrakt osztály között?
Nincs különbség, de interfészekkel szebb megvalósítás érhető el
Egy interfészből tetszőleges számú példány létrehozható, de absztrakt osztályból csak egy.
Egy interfészből tetszőleges számú példány létrehozható, de absztrakt osztályból egy sem.
Interfész csak szignatúrát definiál, az absztrakt osztály megvalósítást is, mely felül nem bírálható
Interfész csak szignatúrát definiál, az absztrakt osztály megvalósítást is, mely felülbírálható

Kérdés: Osztály diagramon mely állítások igazak az aktív osztályokra?
Önmaguktól aktiválódnak, nem kell hozzá külső esemény
Felhasználói akcióra aktiválódnak, ez a külső esemény
<<activeclass>> sztereotípiával hangsúlyozható
<<controller>> sztereotípia kötelező ilyenkor

Kérdés: Osztálydiagramok esetén mi igaz a taxonómiára?
Csak asszociáció tüntethető fel
Csak kompozíció tüntethető fel
Csak öröklődési hierarchia tüntethető fel
Csak olyan osztályok szerepelhetnek rajta, melyek példányosíthatóak

Kérdés: Mit jelent: powertype?
Olyan típusok, melyekben a vezérlés lényegi része található
Olyan osztályok, melyek példányai egyszerre alosztályok is
Olyan osztályok, melyek az öröklési hierarchia tetején találhatóak

Kérdés: UML osztálydiagram esetén melyek a kompozíció hierarchia jellemzői?
Csak osztályok tüntethetők fel, kapcsolatokkal
Csak statikus osztályok tüntethetők fel
Csak osztályok és use casek tüntethetők fel

Kérdés: UML osztálydiagram. Jelölje az igaz állításokat.
Statikus osztály nem példányosítható
Statikus osztály csak tervezési minta szerint példányosítható
Statikus osztály csak egy példánnyal rendelkezhet

Kérdés: Mely állítások igazak? Az UML <<snapshot>> sztereotípiára...
... kizárólag osztálydiagramon használható
... osztály kötelező példányosítását jelöli
... osztály opcionális példányosítását jelöli
... osztály objektumára példát mutat

Kérdés: Mely állítások igazak?
Az ER diagram kinyerhető egy jól elkészített osztálydiagramból

Az EK diagram kinyerhető egy jól elkészített osztálydiagramból

Az ER diagramon meg fog jelenni valamennyi <<entity>> sztereotípiájú osztály

Az ER diagramon meg fog jelenni valamennyi <<er>> sztereotípiájú osztály

Kérdés: Függőségi ciklus hogyan oldható fel

Legtöbbször a kód megduplázásával

A ciklusban lévő csomagok közös csomagba helyezésével

Csomagimporttal

Kérdés: Csomagdiagramon mit jelent a {global} kibővítés?

A csomag elemei mindenhol elérhetőek

A csomag elemei titkosítottak

A csomag elemei nemzetközi használatra készültek

Kérdés: Csomagdiagramon mit jelent a {abstract} kibővítés?

Még meg nem valósított csomagot

Olyan csomagot, melynek nem lehetnek példányai

Kérdés: A függőség tranzitív?

Igen

Nem

Kérdés: Mit jelent a tranzitivitás függőségek esetén?

A függőségi láncok tiltását

A függőség továbbterjedését

Azt, hogy a függő osztályok átlátszóak

Kérdés: Mit értünk függőség alatt? Jelölje az igaz állításokat!

Két tetszőleges elem között akkor létezik függőség, ha az egyik elem specifikációjában, megvalósításában történő változtatás változást okozhat a másik elem specifikációjában, megvalósításában.

Az egyik elem megváltoztatása kihat a másik elemre.

Az egyik elem törlésével a tartalmazott elem is törlésre kerül

Kérdés: Csomagdiagramok. Jelölje az igaz állításokat!

A függőségi kapcsolatok, és a csoportosítás osztályokra vonatkozik

Az UML-ben a leginkább elterjedt rendezési elv a függőség

A fő rajzelem a mappa

A másodlagos rajzelem a use case

A fő sztereotípiát az <<import>>

A fő sztereotípiát az <<uses>>

Kérdés: Mely állítások igazak az interakciós diagramokra?

Az interakciós diagram üzenetváltásokat ábrázol.

Az interakciós diagramon a kölcsönhatásban partner lehet egy osztály

Az interakciós diagramon a kölcsönhatásban partner lehet egy use case

Az interakciós diagramon a kölcsönhatásban partner lehet egy interfész

Az interakciós diagramon a kölcsönhatásban partner lehet egy komponens

Kérdés: Az alábbiak közül melyek tartoznak az interakciós diagramok közé?

Állapotgép

Idődiagram

Állapot átmenet diagram

Kommunikációs diagram

Szekvencia diagram

Kérdés: Mely állítások igazak az interakciós diagramokra?

Használható követelmények leírására

Használható tesztesetek leírására

Elvárt viselkedés specifikálható vele

Használható elvárt rendszer felhasználói felület definiálására

Használható létező rendszer folyamatainak vizsgálatára

Használható létező rendszer folyamatainak dokumentálására

Használható létező rendszer folyamatainak vizualizálására

Kérdés: Interakciós diagramok közül a szekvencia diagram mivel nyújt többet egy kommunikációs diagramnál?

Az alapvető különbség az időzítési megkötések feltüntetésében van.

Nincs különbség, azonos információtartalommal bíró ábrákról van szó.

Kérdés: Jelölje az igaz állításokat!

Szekvencia diagram esetén az aktivitási sáv és az életvonal egyszerre kezdődhet.

Szekvencia diagram esetén az aktivitási sáv kezdete minden esetben megelőzi az életvonal kezdetét, jelezve az inicializálást.

Szekvencia diagram esetén az életvonal végével az aktivitási sávnak is végződnie kell.

Szekvencia diagram esetén az életvonal vége az objektum megszűntét jelöli.

Szekvencia diagram esetén az aktivitási sáv széles sávval van jelölve.

Szekvencia diagram esetén az aktivitási sávban lehetnek szünetek, kimaradások.

Szekvencia diagram esetén az életvonalban lehetnek szünetek, kimaradások

Kérdés: Jelölje az igaz állításokat!

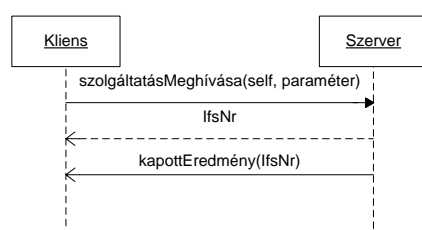
Egy szekvencia diagramon több életvonal lehet.

Egy szekvencia diagramon csak egy aktivitási sáv lehet.

Egy kommunikációs diagramon sok aktivitási sáv is lehet.

Az idődiagramon feltüntethetők az objektum állapotai

Kérdés: Mit lát az ábrán?





Kommunikációs diagramot, amin egy kliens szolgáltatást hív meg egy szerveren, az visszaigazolja, majd az a kapott eredményt visszaadja.

Szekvencia diagramot.

Idődiagramot, de mivel hiányzik az időzítés megkötése, ezért csak az alábbi olvasható ki: egy kliens szolgáltatást hív meg egy szerveren, az visszaigazolja, majd az a kapott eredményt visszaadja

Kérdés: Jelölje az igaz állításokat!

Szekvencia diagramon a kommunikációs partnerek térbeli elhelyezkedése nem számít

Kommunikációs diagramon a kommunikációs partnerek térbeli elhelyezkedése számít

Idődiagramon a kommunikációs partnerek térbeli elhelyezkedése nem számít

Kérdés: Jelölje az igaz állításokat!

Szekvencia diagramon az időt fentről lefelé kell értelmezni.

Kommunikációs diagramon az időt balról jobbra kell értelmezni.

Idődiagramon az időt balról jobbra kell értelmezni.

Kommunikációs diagramon az időt nem értelmezzük.

Szekvencia diagramon az idő nincs értelmezve.

Kérdés: Jelölje az igaz állításokat!

Szekvencia diagramon minden partnerhez fog életvonal tartozni.

Kommunikációs diagramon minden partnerhez fog életvonal tartozni.

Kérdés: Jelölje az igaz állításokat!

Szekvencia diagramok használata ott előnyösebb, ahol kevés partner között, de bonyolult minta szerint történik a kommunikáció.

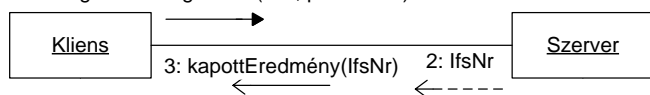
Kommunikációs diagramok használata ott előnyösebb, ahol kevés partner között, de bonyolult minta szerint történik a kommunikáció.

Kommunikációs diagramok használata ott előnyösebb, ahol sok partner között, bonyolult minta szerint történik a nagyszámú üzenetváltás.

Kommunikációs diagramok használata ott előnyösebb, ahol sok partner között, inkább egyszerű minta szerint történik a kommunikáció.

Kérdés: Jelölje az ábrával kapcsolatos igaz állításokat!

1: szolgáltatásMeghívása(self, paraméter)



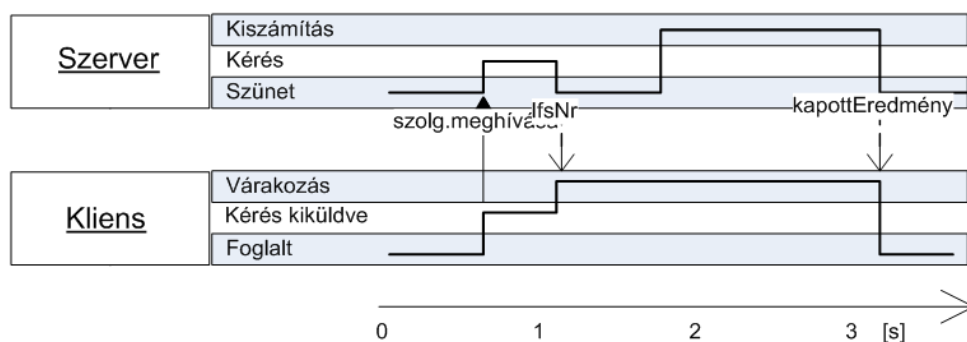
Kommunikációs diagram látható, melyen amin egy kliens szolgáltatást hív meg egy szerveren, az visszaigazolja, majd az a kapott eredményt visszaadja.

Kommunikációs diagram látható, melyen amin egy kliens szolgáltatást hív meg egy szerveren, majd az a kapott eredményt visszaadja.

Hiányos idő diagram látható, melyen amin egy kliens szolgáltatást hív meg egy szerveren, az visszaigazolja, majd az a kapott eredményt visszaadja.

Hiányos szekvencia diagram látható, melyen amin egy kliens szolgáltatást hív meg egy szerveren, az visszaigazolja, majd az a kapott eredményt visszaadja.

Kérdés: Az alábbi ábra alapján jelölje az igaz állításokat!



Lesz olyan szerverre vonatkozó állapotgép, ahol az egyik állapot „várakozás” lesz.

Lesz olyan kliensre vonatkozó állapotgép, ahol az egyik állapot „várakozás” lesz, ebből csak kivezető állapotátmenet lesz.

Lesz olyan kliensre vonatkozó állapotgép, ahol az egyik állapot „várakozás” lesz, ebből több kivezető állapotátmenet lehet.

Lesz olyan szerverre vonatkozó állapotgép, ahol a „Kérés” és „Szünet” állapotok oda-vissza járhatóak

Lesz olyan szerverre vonatkozó állapotgép, ahol a „kiszámítás” állapot elhagyásához olyan eseményre van szükség, mely kapott eredményt küld

Kommunikációs diagramot látunk, időzírtési megköötésekkel.

Egyik állítás sem igaz.

Kérdés: Hogyan kell értelmezni a kontextus interakciós ábrákat?

Olyan idődiagram, melyen környezeti előírások szerepelnek

Olyan szekvencia diagram, melyen nem funkcionális megkötések szerepelnek.

Olyan kommunikációs diagram, melyen környezeti előírások szerepelnek.

Kérdés: Szekvencia diagram esetén hogyan kell értelmezni a STRICT operátort?

A halmazban lévő operandusok szigorú sorrendjét írja elő

Szigorú minőségi követelményeket ír le.

Csak akkor jöhet a következő, ha az előző teljes mértékben befejeződött

Csak kezdési sorrendet szab meg, a befejezés a futásidő függvénye.

Kérdés: Szekvencia diagram esetén hogyan kell értelmezni a REF operátort?

Szigorú minőségi követelményeket ír le.

Csak akkor jöhet a következő, ha az előző teljes mértékben befejeződött

Hivatkozás másik diagramra

Azt szabja meg, hogy a halmazban lévő műveleteknek le kell futnia, opcionálisan.

Kérdés: Szekvencia diagram esetén hogyan kell értelmezni a BRK operátort?

Szigorú minőségi követelményeket ír le.

A megszakítás esetén bekövetkező működést írja le.

Hivatkozás másik diagramra

Tetszőlegesen felhasználható rövidítés, meg kell nézni az UML Modeling Guidelines dokumentumban, hogy aktuálisan mire van használva.

Kérdés: Szekvencia diagram esetén hogyan kell értelmezni a ALT operátort?

Egymást kizáró működéseket ír le.

Csak akkor jöhet a következő, ha az előző teljes mértékben befejeződött és megvan a számítási érték.

A párhuzamosítási alternatívákat jelöli

Kérdés: Szekvencia diagram esetén hogyan kell értelmezni a LOOP operátort?

Egymást kizáró működéseket ír le.

Csak akkor jöhet a következő, ha az előző teljes mértékben befejeződött és megvan a számítási érték.

A párhuzamosítási alternatívákat jelöli
Ismétlődés szemléltetéséhez használható
STRICT és ALT használatát váltja ki.

Kérdés: Szekvencia diagram esetén hogyan kell értelmezni a SEQ operátort?
Egymást kizáró működéseket ír le.
Hasonlít a STRICT-hez, de itt a sorrendiség csak indításkor fontos
A párhuzamosítási alternatívákat jelöli
Ismétlődés szemléltetéséhez használható
STRICT szinonimája, csak ezt általában UML2-ben használják.
Sorba kapcsolja az interakciókat

Kérdés: Szekvencia diagram esetén hogyan kell értelmezni a PAR operátort?
Egymást kizáró működéseket ír le.
Hasonlít a STRICT-hez, de itt a sorrendiség csak indításkor fontos
A párhuzamosan indítandó elemeket jelöli
Ismétlődés szemléltetéséhez használható
STRICT szinonimája, csak ezt általában UML2-ben használják.
Egyidejűséget jelöl

Kérdés: Szekvencia diagram esetén hogyan kell értelmezni a REGION operátort?
Egymást kizáró működéseket ír le.
Hasonlít a STRICT-hez, de itt a sorrendiség csak indításkor fontos
A párhuzamosan indítandó elemeket jelöli
Ismétlődés szemléltetéséhez használható
PAR esetén a kötelezően egyben végrehajtandó részt jelöli
Egyidejűséget jelöl

Kérdés: Jelölje az igaz állításokat!
--------------------------------------

Szekvencia diagramon feltüntethető az objektum aktuális állapota, téglalapba írva.

Szekvencia diagramon feltüntethető az objektum aktuális állapota, lekerekített téglalapba írva.

Szekvencia diagramon feltüntethető az objektum aktuális állapota, ellipszisbe írva.

Kérdés: Jelölje az igaz állításokat!

Az interakciós áttekintés olyan tevékenységdiagram, melyen szekvencia diagramokat hivatkozhatunk.

Az interakciós áttekintés olyan osztálydiagram, melyen szekvencia diagramokat hivatkozhatunk.

Az interakciós áttekintés olyan szekvencia diagram, melyen kommunikációs diagramokat hivatkozhatunk.

Az interakciós áttekintés olyan szekvencia diagram, melyen aktivitási diagramokat hivatkozhatunk.

Kérdés: Jelölje az igaz állításokat!

Osztálydiagramon az osztály rajzelemben feltüntethetőek objektum életciklus hivatkozások.

Osztálydiagramon az osztály rajzelemben feltüntethetőek use case hivatkozások.

Az objektum életciklus diagram bemutatja, hogy milyen műveletek hatására melyik állapotból, melyik állapotba kerül egy példány.

Az állapotgép tulajdonképpen nem más, mint egy use-case és metódus-megvalósítás összerendelés

Kérdés: Jelölje az igaz állításokat!

Állapotgépen az összetett állapot jele egy lekerekített téglalap, benne két kisebb jellel.

Állapotgépen az összetett állapot jele lehet egy három részre osztott lekerekített téglalap.

Állapotgépen az összetett állapot jele lehet egy két részre osztott lekerekített téglalap.

Kérdés: Jelölje az igaz állításokat!

Állapotgépen az összetett állapot jele lehet egy két részre osztott lekerekített téglalap – fent az állapot neve, lent tevékenységdiagram részlettel

Állapotgépen az összetett állapot kibontható és több alállapotot tartalmaz, állapotátmenetekkel

Állapotgépen az összetett állapot kibontható és több alállapotot tartalmaz, állapotátmenetek nélkül.

Kérdés: Jelölje az igaz állításokat!

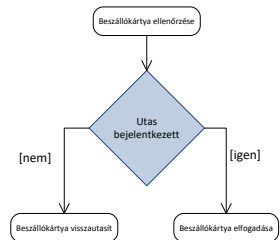
Jelölése téglalap, benne a választás, és a kimeneteken szögletes zárójelben a feltételek.

Állapotgépen az esetválasztó csúcs csak egymást kizáró kimeneteket tartalmazhat.

Jelölése rombusz, benne a választás, és a kimeneteken szögletes zárójelben a feltételek.

Állapotgépen az esetválasztó csúcs nem csak egymást kizáró kimeneteket tartalmazhat, valamint párhuzamos működés esetén jelezheti a párhuzamos szakasz kezdetét.

Kérdés: Az ábra alapján jelölje az igaz állításokat!

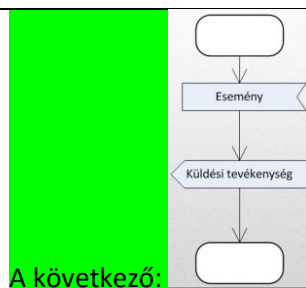
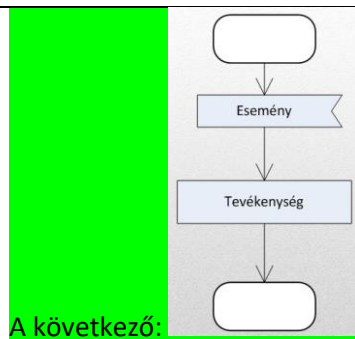


Az ábrán egy állapotgép részlet látható, mely adott osztály működését írja le.

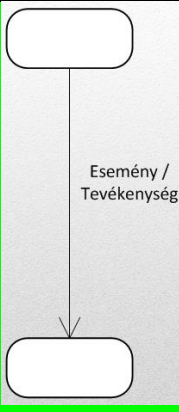
Az ábrán egy forgatókönyvhöz tartozó aktivitás diagram részlet látható, mely alapján kialakítható a csomagdiagram.

Kérdés: Jelölje, mely ábrázolási módok szabványosak UML2-ben, állapotautomatákra!

Egyik sem



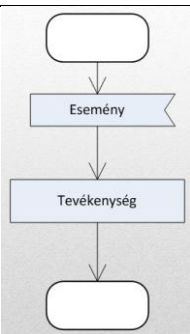
A következő:



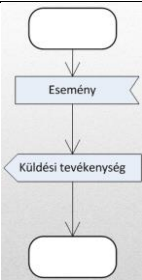
Kérdés: Jelölje, mely ábrázolási módok szabványosak UML2-ben, aktivitási diagramokra!

Egyik sem

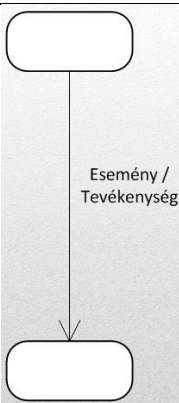
A következő:



A következő:



A következő:



Kérdés: Állapotgépekkel kapcsolatban melyik állítás(ok) igaz?

Egyszerű történeti állapot esetén (Közvetlen tartalmazó állapotba érkezéskor a közvetlen tartalmazott régióban elsőként) a történeti állapot rákövetkező állapotába kerül a vezérlés, amennyiben a történeti állapot üres.

Ha nem üres, akkor azt az állapotot veszi fel újra az automata, melyben akkor volt, amikor a közvetlen tartalmazó állapot elhagyásra került.

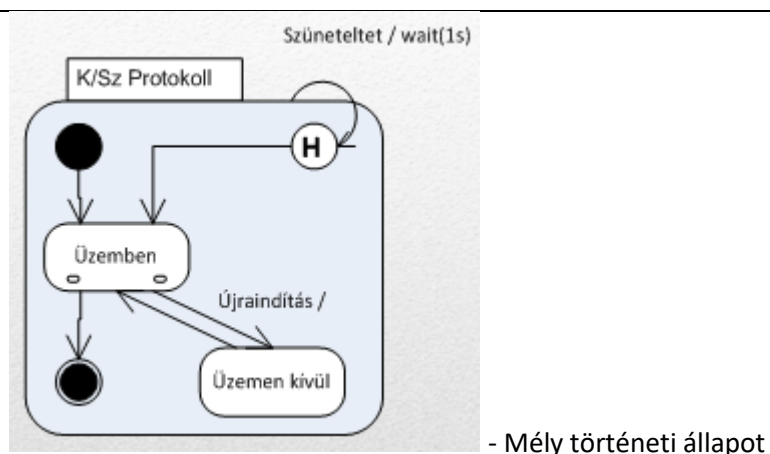
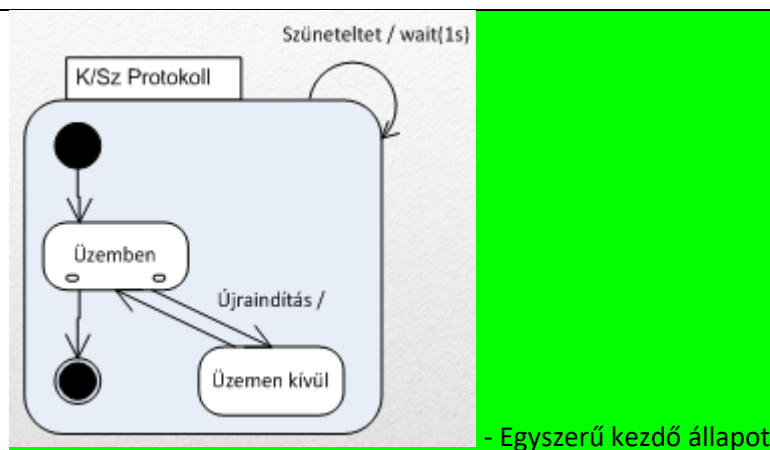
Egyszerű történeti állapot esetén (Közvetlen tartalmazó állapotba érkezéskor a közvetlen tartalmazott régióban elsőként) a történeti állapotba kerül a vezérlés, amennyiben a történeti állapot üres.

Ha nem üres, akkor azt az állapotot veszi fel újra az automata, melyben akkor volt, amikor a közvetlen tartalmazó állapot elhagyásra került.

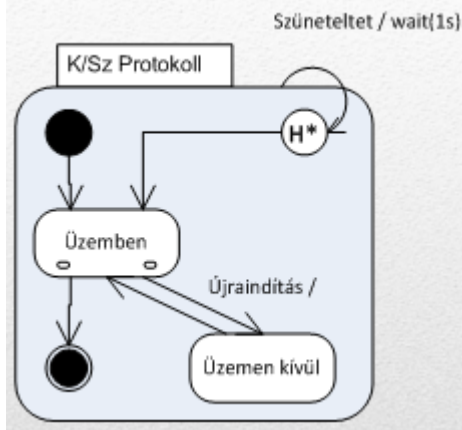
Egyszerű történeti állapot esetén (Közvetlen tartalmazó állapotba érkezéskor a közvetlen tartalmazott régióban elsőként) a történeti állapot rákövetkező állapotába kerül a vezérlés, amennyiben a történeti állapot üres.

Ha nem üres, akkor azt az állapotot veszi fel újra az automata, melyben akkor volt, amikor a közvetlen tartalmazó állapot elhagyásra került. Az állapotkonfiguráció is visszaállításra kerül.

Kérdés: Állatgépek kapcsán melyik összerendelés(ek) helyes?







- Egyszerű történeti állapot

Kérdés: Igaz-e az alábbi állítás:

Mély történeti állapot estén (Közvetlen tartalmazó állapotba érkezéskor a közvetlen tartalmazott régióban elsőként) a történeti állapot rákövetkező állapotába kerül a vezérlés, amennyiben a történeti állapot üres.

Ha nem üres, akkor azt az állapotot veszi fel újra az automata, melyben akkor volt, amikor a közvetlen tartalmazó állapot elhagyásra került, de nem csak az állapot, hanem az állapotkonfiguráció kerül visszaállításra.

Igaz

Hamis

Kérdés: Igaz-e az alábbi állítás:

Egyszerű történeti állapot estén (Közvetlen tartalmazó állapotba érkezéskor a közvetlen tartalmazott régióban elsőként) a történeti állapot rákövetkező állapotába kerül a vezérlés, amennyiben a történeti állapot üres.

Ha nem üres, akkor azt az állapotot veszi fel újra az automata, melyben akkor volt, amikor a közvetlen tartalmazó állapot elhagyásra került, de nem csak az állapot, hanem az állapotkonfiguráció kerül visszaállításra.

Igaz

Hamis

Kérdés: Jelölje az igaz állításokat!

A dialóguslefutás az állapotgép egyik fajtája, de állapotokként a képernyő állapotait kell feltüntetni.

A dialóguslefutás az tevékenységdiagram egyik fajtája, de tevékenységekként a képernyő helyzetét kell feltüntetni.

A dialóguslefutás az objektumdiagram egyik fajtája, de objektumok helyett a képernyőket kell feltüntetni.

Kérdés: Jelölje az igaz állításokat!

Egy UML-modell összefüggő (konzisztens), ha use case diagramjain nincs sem olyan use case, sem olyan aktor, mely nem vesz részt interakcióban.

Egy UML-modell összefüggő (konzisztens), ha osztálydiagramján nincs elszigetelt osztály.

Egy UML-modell összefüggő (konzisztens), ha létezik hozzá egy bizonyos lehetséges működőképes implementáció.

Kérdés: Jelölje az igaz állításokat!

Egy inkonzisztens specifikáció azzal jár, hogy nem lehet meghatározni a use casek közötti összefüggéseket.

Egy inkonzisztens specifikáció azzal jár, hogy nem lehet belőle olyan szoftvert fejleszteni, ami működőképes lenne.

Egy inkonzisztens specifikáció azzal jár, hogy lesznek olyan osztályok, melyek ugyan meg vannak valósítva, de használatban nincsenek.

Kérdés: Jelölje az igaz állításokat!

Egy UML tervezési modellt teljesnek nevezünk, ha az általa leírt szoftver-rendszer állapotai és viselkedési módjai mindegyik lehetséges működési esetre és feltételre specifikálva lettek.

Egy UML tervezési modellt teljesnek nevezünk, ha abban minden – szakterület által megkívánt – aktor megtalálható.

Egy UML tervezési modellt teljesnek nevezünk, ha az általa leírt use casek és forgatókönyvek mindegyik lehetséges forgatókönyvek specifikálva lettek.

Kérdés: Jelölje az igaz állításokat!

Jelenleg úgy a teljesség, mint a konzisztencia megléte bizonyítható emulációs szoftverek segítségével.

Jelenleg nincs megoldva még a teljesség bizonyításának vagy cáfolásának feladata.

Jelenleg ugyan rendelkezésre áll olyan módszer, ill. a módszert megvalósító számítógépes eszköz, ami egy szoftver-terv UML-diagramjai közötti összhang bizonyítására, ill. cáfolására alkalmas lenne, ám ennek használata nem terjedt el.

Jelenleg nem áll rendelkezésre olyan módszer, ill. a módszert megvalósító számítógépes eszköz, ami egy szoftver-terv UML-diagramjai közötti összhang bizonyítására, ill. cáfolására lenne alkalmas.

Kérdés: Jelölje az igaz állításokat!

A komponens diagram a komponensekből felépülő szoftver-rendszer struktúráját vázolja fel.

A komponens adott modellelemek (osztályok, csomagok) fizikai egysége.

A komponens diagram jól modellezi a szoftver-komponensek egymáshoz való viszonyát, kommunikációját.

A komponens adott modellelemek (use case és forgatókönyv) fizikai egysége.

Kérdés: Jelölje az igaz állításokat!

UML szimbóluma egy téglalap, amelynek bal oldalát két kisebb téglalap metszi.

Egy komponensben csak egy implementációs osztály valósulhat meg, vagy számos tervezési osztály.

UML szimbóluma egy téglalap, amelynek jobb felső sarkát körbe írt paraméter jellemezheti..

Egy komponensben számos implementációs osztály valósulhat meg.

Kérdés: Jelölje az igaz állításokat!

Komponens diagramon a <<executable>> sztereotípa használható komponensek jellemzésére.

Komponens diagramon a <<trace>> sztereotípa használható komponensek jellemzésére.

Komponens diagramon a <<implements>> sztereotípa használható komponensek jellemzésére.

Komponens diagramon a <<snapshot>> sztereotípa használható komponensek jellemzésére.

Komponens diagramon a <<file>> sztereotípa használható komponensek jellemzésére.

Kérdés: Jelölje az igaz állításokat!

Komponens diagramon a <<file>> sztereotípa használható komponensek jellemzésére.

Komponens diagramon a <<instanceOf>> sztereotípa használható komponensek jellemzésére.

Komponens diagramon a <<library>> sztereotípa használható komponensek jellemzésére.

Komponens diagramon a <<property>> sztereotípa használható komponensek jellemzésére.

Komponens diagramon a <<document>> sztereotípa használható komponensek jellemzésére.

Kérdés: Jelölje az igaz állításokat!

A telepítési diagram minden egyes csomópontja (node) valamilyen számítási feldolgozási egységet képvisel, egy szoftver vagy egy hardver elemet.

A telepítési és komponens diagramok nem kombinálhatóak közös ábrába.

A telepítési diagram minden egyes csomópontja (node) valamilyen számítási feldolgozási egységet képvisel, egy szoftver vagy egy hardver elemet. A hardver lehet egy kisebb egység, de lehet egy teljes számítógép is.

A telepítési és komponens diagramok nem kombinálhatóak közös ábrába.

Kérdés: Jelölje az igaz állításokat!

Az UML-specifikáció részét képező OCL nyelv megszorítások definiálására alkalmas.

Az UML-specifikáció részét képező OCL nyelv pszeudo-kód és ábrarészletek definiálására alkalmas.

A <<boundary>> sztereotípa osztályokra alkalmazható.

A <<boundary>> sztereotípa képernyőrészletekre alkalmazható.

OO tervezésben egy CRC-kártya osztályokhoz felelősségeket határozhat meg.

OO tervezésben egy CRC-kártya fejlesztőkhöz felelősségeket határozhat meg.

Kérdés: Jelölje az igaz állításokat!

Több objektum együttes működésének leírására az állapotgép alkalmas.

Függőség két osztály között akkor áll fenn, ha van közöttük közös aktor.

Függőség két osztály között akkor áll fenn, ha egyik specifikációjának megváltozása kihathat a másik osztály specifikációjára.

Az absztrakt osztály csak limitált számú példánnyal rendelkezhet.

Kérdés: Jelölje az igaz állításokat!

Az absztrakt osztály nem példányosítható.

Az absztrakt osztály nem példányosítható, de öröklöthető belőle másik osztály.

Az absztrakt osztály nem példányosítható és öröklésben is csak egy leszármazottja lehet.

Kérdés: Jelölje az igaz állításokat!

Többszintű öröklődés esetén van olyan osztály, melynek őse és leszármazottja is van.

Többszörös öröklődés esetén van olyan osztály, melynek őse és leszármazottja is van.

Többszörös öröklődés esetén lehet olyan osztály, melynek több őse és több leszármazottja van.

Többszörös öröklődés esetén lehet olyan osztály, melynek több őse és egy leszármazottja van.

Kérdés: Jelölje az igaz állításokat!
Egy adott objektum dinamikus működésének leírására az osztálydiagram szolgál.
Egy adott objektum dinamikus működésének leírására az objektumdiagram szolgál.
Egy adott objektum dinamikus működésének leírására az állapot-átmenet diagram szolgál.
Egy adott objektum dinamikus működésének leírására a tevékenységdiagram szolgál.

Kérdés: Alapvetően melyik UML diagramon lehetséges üzenetek feltüntetése?
Use case diagram
Szekvencia diagram
Állapotgép
Osztálydiagram

Kérdés: Alapvetően melyik diagramon lehet események bekövetkezését jelölni?
Use case diagram
Szekvencia diagram
Állapot-átmenet diagram
Osztálydiagram

Kérdés: Jelölje az osztálydiagramon osztályokra alkalmazható sztereotípiákat!
<<implements>>
<<uses>>
<<entity>>
<<boundary>>

Kérdés: Jelölje az igaz állításokat!
Refactoring során elképzelhető, hogy új metódusok kerülnek egy osztályba.
Refactoring során a funkcionalitás nem változik meg.

Kérdés: Mit jelent az emlékező állapot?
Olyan állapot, amelyik emlékszik arra, hogy melyik állapotról származott a vezérjel, amikor odaért
Olyan állapot, amelyik emlékszik arra, hogy melyik alállapotról terminált, és képes arra, hogy az állapotba való újabb belépéskor ugyanabba az állapotba kerüljön.
Olyan állapot, amelyik emlékszik arra, hogy a vezérjel hányszor volt már jelen ebben az állapotban, illetve igény esetén mérhető az állapotban töltött idő hossza is

Kérdés: Mi a vezérlőautomata?
A vezérlőautomata olyan telepítési diagram, mely műszaki eszközt, rendszert vezérel, vagy ír le
A vezérlőautomata olyan osztálydiagram, mely műszaki eszközt, rendszert vezérel, vagy ír le
A vezérlőautomata olyan állapotgép, mely műszaki eszközt, rendszert vezérel, vagy ír le

Kérdés: Mi a dialóguslefutás?
Olyan állapotautomata, mely a képernyő állapotait modellezi
Olyan szek
A vezérlőautomata olyan állapotgép, mely műszaki eszközt, rendszert vezérel, vagy ír le