

VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

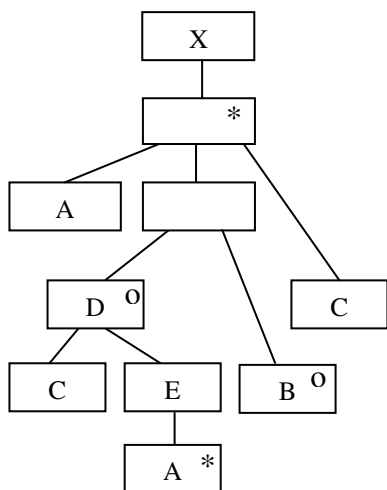
c. tárgyból

2014. január 7.

Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.

A tesztkérdésekre adott rossz válasz esetében pontot veszít, de feladatonként a total pontszám ≥ 0

1. Egészítse ki az **állapottáblát** úgy, hogy az ELH-nak megfelelő szerkezetet írja le! A kiegészítés során **csak az állapottábla üres blokkjaiba írhat**, bármi egyebet az ELH-n vagy az állapottáblán változtatni tilos! Az induló állapot legyen az ① ! (5 pont)



	A	B	C	
①	②			
②		④	③	
③	③		①	
④			①	

Az alábbiak közül mely deklarációk hiányoznak az ELH-t leíró DTD-ből? (3 pont)

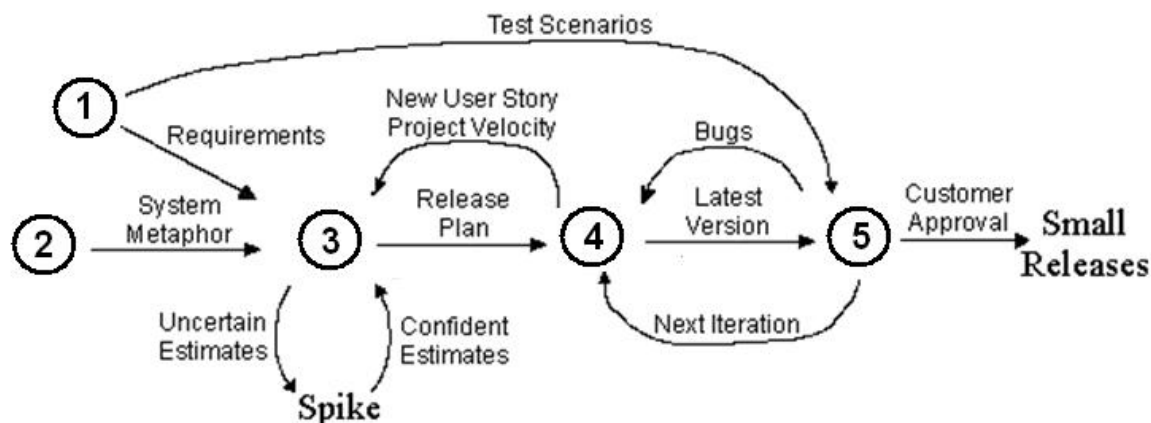
- X (A, Q, C) *
- Q (D|B)
- D (C, E)
- D (A) +
- E (A) *
- Y (C, A*|B?)
- Y ((C, A) *|B)
- X (A, (D|B), C) *
- D (C, E, A*|B)
- X (A, D?, C) *

```
<?xml version="1.0"
encoding="ISO-8859-1"?>
<!DOCTYPE X [
  <!ELEMENT A (#PCDATA)>
  <!ELEMENT B (#PCDATA)>
  <!ELEMENT C (#PCDATA)>
]>
```

2. JUnit-tal tesztelni akarjuk a nem kezelt kivételt. Az alábbi lehetőségek közül mely tesztmetódusok futnak le sikeresen, ha a program nem kezelt ArithmeticException (pl. nullával osztáskor) kivételt dob? Jelölje X-szel a helyes választ! (2 pont)

	@Test(throwable=ArithmeticException) public void test() {program, ami kivételt dob}
	@Test public void test()throws ArithmeticException {program, ami kivételt dob}
X	@Test(expected=ArithmeticException.class) public void test() {program, ami kivételt dob}
	JUnit-tal nem kezelt kivétel nem tesztelhető

3. Az Extrém Programozás (XP) projektet írja le az alábbi (hiányos) ábra. Adja meg azoknak az elemeknek a számát és nevét, amelyek a Rational Unified Process (RUP) alapelveihez közvetlenül kapcsolódnak (3 pont)



száma	XP elem neve	RUP alapelv
1	User stories	Use case vezérelt
2	Architectural spike	Architektúra központú
4	Iteration	Iteratív, inkrementális

4. Jelölje (karikázza be) az állítások igazságtartalmát, ha feltesszük, hogy szabványos Java nyelvet használunk! (9 pont)

- I H *final* osztálynak nem lehet *abstract* metódusa.
- I H minden generikus osztály használható *Object* paraméterezéssel.
- I H generikus osztály paramétere nem lehet primitív típus.
- I H primitív típus tömbje is a primitív típusok közé számít.
- I H ha egy szál véget ért, akkor *start()* metódushívással újraindítható.
- I H egy szál csak akkor hajthat végre *wait()* metódushívást, ha a hívott objektum monitorában tartózkodik.
- I H ha egy szál *notify()* hívás hatására hagyja el a *WAIT* állapotot, akkor *RUNNABLE* állapotba kerül.
- I H *String* objektum tartalma bármikor megváltoztatható.
- I H egy osztály statikus metódusát csak statikus metódusból lehet meghívni.

5. Ha metódus meghívásakor a kliens oldalon hibázunk, az mely szerződéses feltételek megsértését jelenti? (2 pont)

előfeltétel (precondition)

6. Rajzoljon UML2 állapotdiagramot a Q osztályhoz! (5 pont)

```

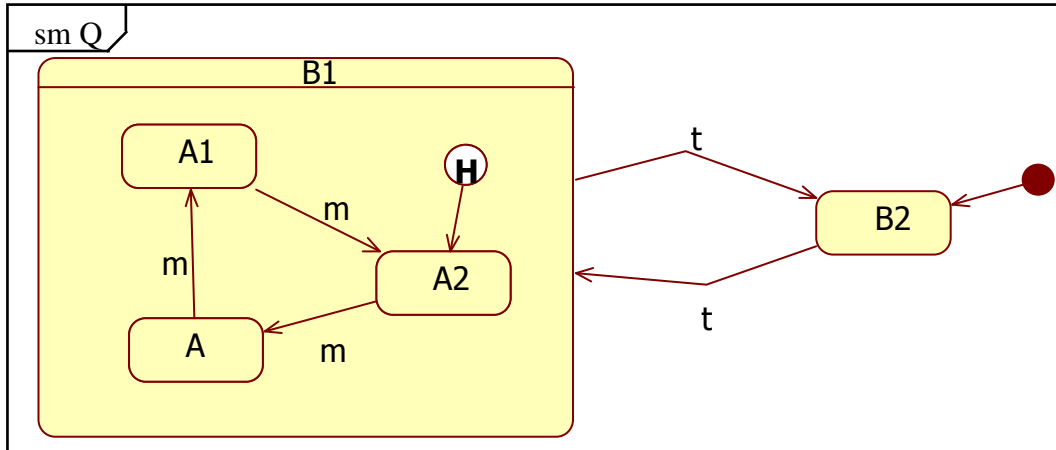
enum AS { A1, A2, A3 }
enum BS { B1, B2 }

class Q {
private AS as = AS.A2;
private BS bs = BS.B2;

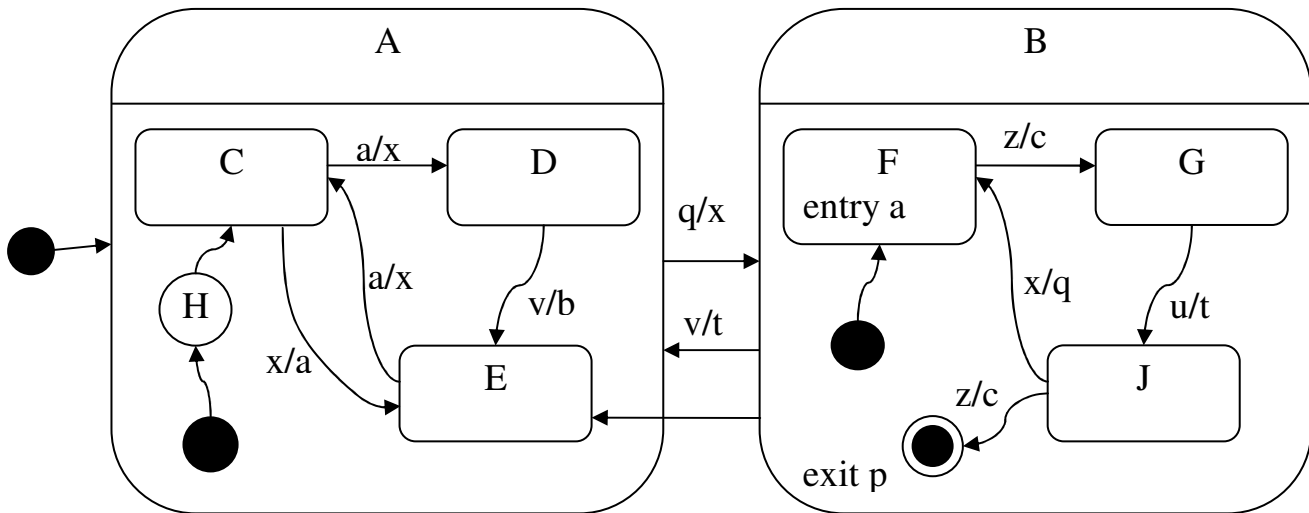
public void t() {
  if (bs == BS.B1) {bs = BS.B2;}
  else {bs = BS.B1;}
}
}

public void m() {
  if (bs == BS.B1) {
    switch (as) {
      case A1: as = AS.A2; break;
      case A2: as = AS.A3; break;
      case A3: as = AS.A1; break;
    }
  }
}
}

```



7. A következő UML2 állapotdiagram alapján minősítse az állításokat! (8 pont)



Igaz	Hamis	Állítás
<input type="checkbox"/>	<input type="checkbox"/>	D állapotból 2 lépésben visszaérhet D-be
<input type="checkbox"/>	<input checked="" type="checkbox"/>	J állapotból „p” esemény hatására E állapotba kerül
<input type="checkbox"/>	<input checked="" type="checkbox"/>	F állapotból „v” esemény hatására H állapotba kerül
<input checked="" type="checkbox"/>	<input type="checkbox"/>	A-ból B-be való váltáskor mindig végrehajtódik az „a” tevékenység

A kezdés után az **x, q, z, u, z** esemény-szekvencia hatására

Igaz	Hamis	Állítás
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Egyszer lefut a „p” tevékenység
<input type="checkbox"/>	<input checked="" type="checkbox"/>	C állapotba kerülünk
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Az E állapotot kétszer érintjük
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Kétszer lefut az „a” tevékenység

8. Az alábbi állapotgéppel specifikált programot az interpreter architektúra szerint kívánjuk megvalósítani.

	A	B	C
D	D/E	D/-	F/G
F	D/-	D/E	D/-

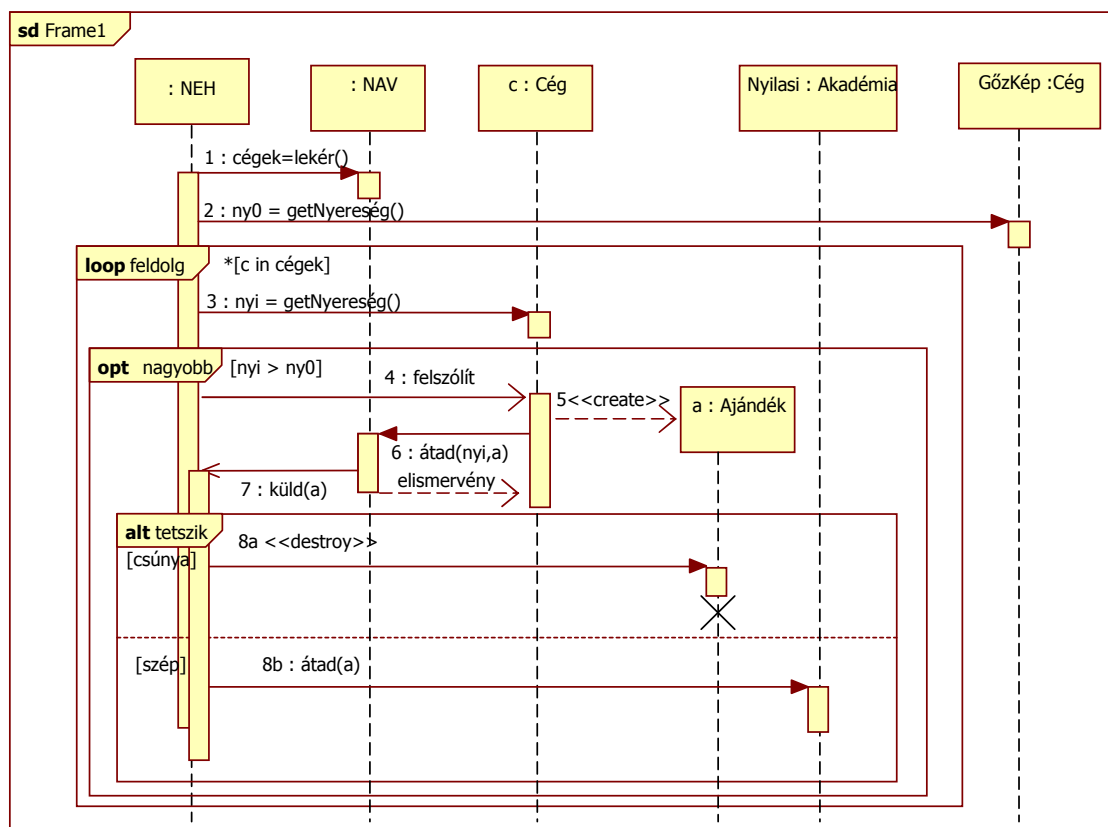
Specifikálja a

“pseudocode” utasításait ! (3 pont): **D/E, D/-, F/G**.....

“control state” lehetséges értékeit ! (2 pont): **[D|F]**.....

9. Készítsen UML2 szekvenciadiagramot az alábbi történet alapján! (8 pont)

A Nemzeti Egyensúly Hivatal (NEH) lekéri a NAV-tól a bekötött pénztárgép cégek listáját. A listán szereplő cégektől elkéri az éves eredményt, és amelyik cégnél a GőzKép ZRt-ét meghaladó nyereség keletkezett, levélben felszólítást küld. A cégnek a felszólítás hatására azonnal kis ajándékot kell készítenie, és az ajándékkal együtt át kell adnia a nyereségét a NAV-nak. A NAV az ajándékot azonnal továbbküldi a NEH-nek, majd a cég felé elismervénnyel nyugtázza az adomány átvételét. Ha az ajándék csúnya, a NEH megsemmisíti, ha szép, akkor a Nyilasi Akadémiának adja át.



Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5