

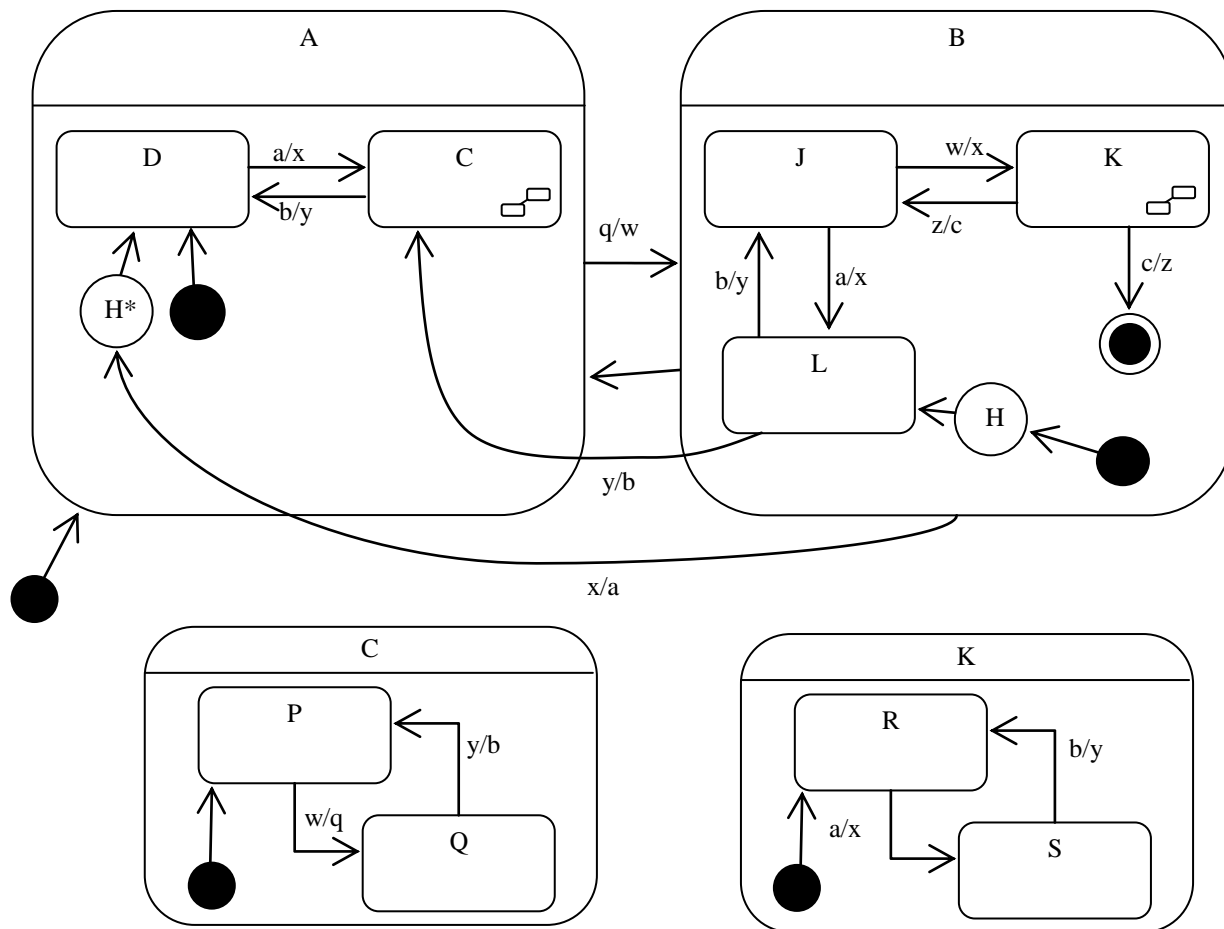
VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból
2010. december 21.

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. A következő UML állapotdiagram alapján minősítse az állításokat! Csak a rubrikába tett jelzést vesszük figyelembe! (8 pont)

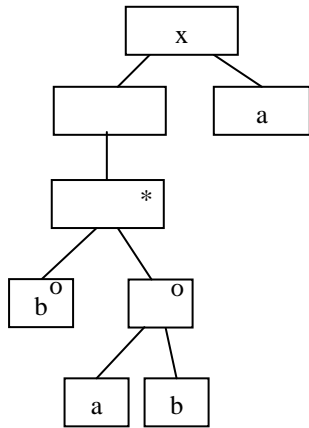


Igaz	Hamis	Állítás
<input type="checkbox"/>	<input type="checkbox"/>	L állapot után közvetlenül következhet Q állapot
<input type="checkbox"/>	<input type="checkbox"/>	C állapotból elérhető egy lépésben S
<input type="checkbox"/>	<input type="checkbox"/>	K állapotból csak „c” és „x” esemény hatására léphetünk át A állapotba
<input type="checkbox"/>	<input type="checkbox"/>	Q állapotból „y” esemény hatására átlépünk D-be
<input type="checkbox"/>	<input type="checkbox"/>	L állapot után csak C és J következhet egy lépésben

A kezdés után az **a, w, q, b, x** esemény-szekvencia hatására

Igaz	Hamis	Állítás
<input type="checkbox"/>	<input type="checkbox"/>	pontosan kétszer fut le a „q” tevékenység
<input type="checkbox"/>	<input type="checkbox"/>	Q állapotba kerülünk
<input type="checkbox"/>	<input type="checkbox"/>	érintettük az K állapotot

2. Adott az alábbi Jackson-ábra.



```
<?xml version="1.0"
encoding="ISO-8859-1"?>
<!DOCTYPE x [
  <!ELEMENT x      ①      >
  <!ELEMENT a      (#PCDATA)>
  <!ELEMENT b      (#PCDATA)>
]>
<x>
  <a>a</a>
  <b>b</b>
  <b>-x</b>
  <b>x</b>
  <a>-x</a>
</x>
```

Jelölje meg, hogy melyik elv jelenik meg a szerkezetben egynél többször ! (2 pont)

- szekvencia (sorrend)
- szelekció (választás)
- iteráció (ismétlődés)
- egyik sem

Az alábbiak közül ① helyén melyik válasz írja le helyesen a Jackson-ábrát ? (4 pont)

- $(a^*, b \mid (a, b))$
- $(((a, b) \mid b)^*, a)$
- $(b^*, (a, b) \mid a)$
- $((b, (a, b)^*), a)$
- $((b \mid (a, b))^*, a)$
- $((b \mid (a, b)^*), a)$
- $(((a, b) \mid b^*), a)$
- egyik sem

A baloldalon álló XML adatszerkezet (2 pont)

- mint XML szerkezet szintaktikai hibás
- megfelel a Jackson-ábrának (érvényes)
- nem felel meg a Jackson-ábrának (érvénytelen)
- érvényessége nem dönthető el

3. Kockázatelemzés (risk analysis) során minden kockázathoz hozzárendeljük (4 pont)

.....

.....

4. Egy osztály megvalósít egy absztrakt adatstruktúrát. Az osztály metódusai az adatstruktúra műveletei. Milyen a metódusok közötti kohézió ? (2 pont)

- funkcionális
- strukturális
- kommunikációs
- procedurális
- temporális
- egyik sem, hanem

Egy egyszerű társalgó (chat) program úgy működik, hogy a bejelentkezett felhasználó által küldött üzeneteket a rendszer minden másik bejelentkezett felhasználónak továbbítja. Milyen a szoftver architektúrája ? (2 pont)

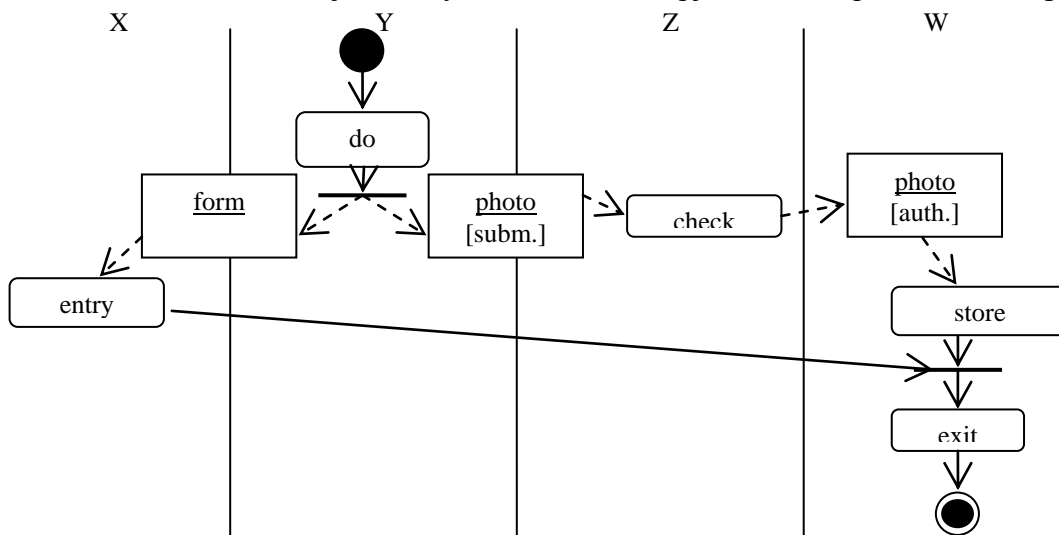
..

A következő feladatokat csak akkor értékeljük, ha az előző lapon szereplő feladatokból minimum 14 pontot ért el.

5. Készítsen UML2 use-case diagramot az alábbi leírás alapján! (7 pont)

A sarokhegesztővel a junior furga buzzogni, és jarmolni tud. A buzzogáshoz be kell kapcsolni a frincolást. Ennek két módja van: a prantolás és a hrufdolás. A herkentyű hibás beállítások esetén frincolás közben brenzel is kicsit. A szenior furga a fentiek mellett a durgolás funkcióhoz is hozzáfér, amihez szintén be kell kapcsolni a fenti frincolás funkciót.

6. Legyen adott az alábbi – object flow-val kiegészített – aktivitás-diagram (activity diagram) ! Rajzolja meg azon objektumok UML2 state-chartját, amelyeknek az ábra alapján több állapota is van! (4 pont)



7. Definiálja, hogy mi a "konfiguráció" a konfiguráció menedzsmentben ! (3 pont)

8. Készítsen UML 2 osztálydiagramot (class diagram) az alábbi leírás alapján! Használja a kövéren szedett kifejezéseket! Ahol lehet, adja meg a paraméterek, attribútumok, stb típusát is! (8 pont)

Az OOniX operációs rendszerben **folymatok**, **fájlok** és hálózati kapcsolatok (**socket**) vannak. A folymatoknak van azonosítója (**pid**), tulajdonosa (**uid**). Egy folymatból a **fork()** metódussal lehet újat létrehozni. Minden folymat ismeri a közvetlen **őst** és a **gyerekeit**. A fájlknak van egy senki más által nem látható **inode** száma, és lekérdezhető a **méretük**. A fájlok többfélék lehetnek: **könyvtárak**, amelyek más fájlokat tartalmazhatnak (a **nevük** alapján), **reguláris** fájlok, amiknek van **típusa**, stb. Minden fájl egy könyvtár része. A folymatok egyformán kezelhetnek socketet és fájlt is, de csak egy közös interfészt (**stream**) látnak belőlük, amiken bájtokat lehet **olvasni** és **írni**. Az ilyen objektumokról a folymatnak van egy listája, aminek az elemeit fájlleíróval (**fd**) azonosítja. A folymatok létrehozásakor egy (a folymatok számára közös) számláló (**lastpid**) növekszik, ez lesz az újonnan létrehozott folymat azonosítója. A folymatoknak más folymatok tudnak üzenni a **signal()** üzenet meghívásával (egy darab egész típusú paramétere van). A lastpid és a signal csak folymatból (és esetleges leszármazottjából) látható.

9. A Rational Unified Process (RUP) követelmény szakaszában milyen típusú use-case-eket készítnk ? Mi ennek a típusú use-case-nek a fő jellemzője ? (4 pont)

Eredmények értékelése:

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