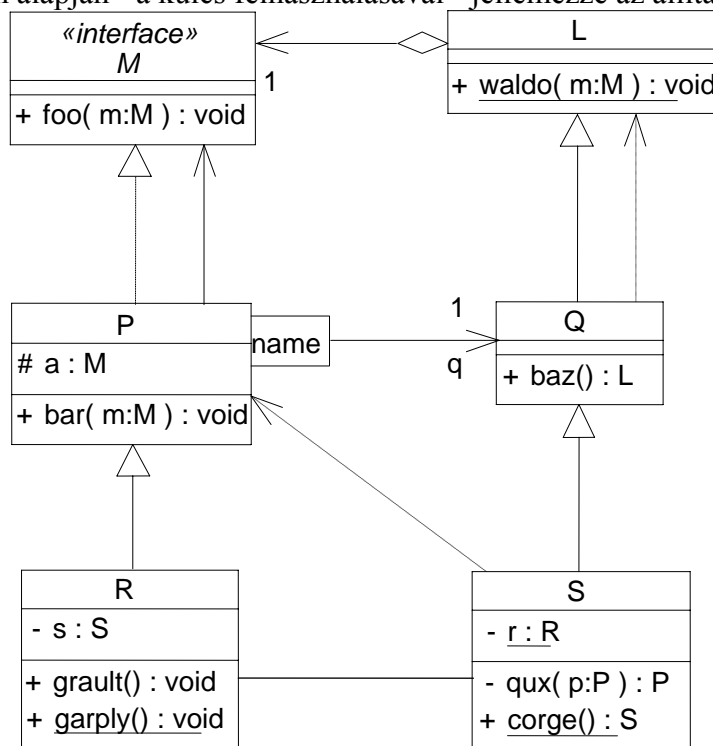


VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból
2011. június 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.

1. Az alábbi UML2 diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)



- | | |
|--|---------|
| A - csak az első tagmondat igaz | (+ -) |
| B - csak a második tagmondat igaz | (- +) |
| C - mindkét tagmondat igaz, de a következtetés hamis | (+ + -) |
| D - mindkét tagmondat igaz és a következtetés is helyes | (+ + +) |
| E - egyik tagmondat sem igaz | (- -) |

R garply metódusa meghívhatja az **s** attribútum **corge** metódusát, mert a **garply** és a **corge** metódus is statikus.

P nem hívhat **Q** osztályon **waldo** függvényt, mert **L waldo** függvénye nem virtuális.

Q függ **M**-től, mert **Q** ismeri **P**-t.

Az **M** típus közvetlenül nem példányosítható, ezért **R grault** metódusa nem hívhatja meg **P a** attribútumának **foo** metódusát.

Egy **P** típusú objektum pontosan egy **Q** típusú objektumot ismer, ezért **P** függ **Q**-tól.

S az **L** leszármazottja, ezért **Q baz** függvénye példányosíthat **S** típusú objektumot.

S qux függvénye nem módosíthatja az **r** attribútum **s** attribútumát, mert **R s** attribútuma privát.

S ismeri **M**-et, mert az **S** osztály **L** őse függ **M**-től.

2. Az E entitás élettörténetét az alábbi állapotábrával írhatjuk le (A a kezdőállapot).
Rajzolja fel az élettörténetet JSD ábrán és szintaxisgráffal! (6 pont)

	x	y	z
A		A	B
B	A	A	

3. Mi a lényege a JSD-ben a “funkció” lépésnek (function step)? (2 pont) Rossz válasz esetén pontot veszít !

- Az entitások funkcióinak specifikálása
- A funkcionális dekompozíció
- Az entitások kibővítése funkció-entitásokkal
- Funkció processzek élettörténetének tervezése
- Funkció processzek felvétele a processzek hálózatába
- Funkció processzek ütemezése

Mi az **előző** lépés? (1 pont)

Mi a **következő** lépés? (1 pont)

4. Az esemény alapú implicit hívást használó (event-based implicit invocation) architektúrának mi a két fő komponense és mi a komponensek szerepe? (3 pont)

5. Sorolja fel a Rational Unified Process (RUP) élekciklus modelljében szereplő „támogató munkafolyamatokat” (supporting workflows)! (3 pont)

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

6. A RUP-ban (Rational Unified Process) alkalmazott használati esetek (use-case-ek) különböző szempontok szerint csoportosíthatóak. Jellemezze a **lényeges** (essential) és a **valóságos** (real) use case-eket! Miben különböznek egymástól? (3 pont)

7. Adott az alábbi XML leírás.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<!DOCTYPE x [  
  <!ELEMENT x (y+, c)>  
  <!ELEMENT y (a | (b+, c?))>  
  <!ELEMENT a (#PCDATA)>  
  <!ELEMENT b (#PCDATA)>  
  <!ELEMENT c (#PCDATA)>  

```

Készítsen a DTD-nek megfelelő érvényes (valid) és szintaktikailag helyes (jól formált) XML adatszerkezetet, amelyben van pontosan két c elem van! (2 pont) Az XML deklaráció (<?xml version="1.0" encoding="ISO-8859-1"?>) nem kell.

8. Készítsen UML2 komponens diagramot az alábbi leírás alapján! (7 pont)

A SpaceFix magyar műholdkarbantartó-modul felületén van fűrés (ami elvárja, hogy valamit fűrészelhessen), antennabemenet és tápbemenet. Ha a modult szétszedjük, akkor azt látjuk, hogy valójában egy flexből és egy IT-modulból áll. A flex biztosítja a fűrészelést és kapja a tápbemenetet, míg az antennabemenet az IT-modulhoz kapcsolódik. A flex olajozónylással és vezérlőgombbal is rendelkezik, az IT-modul ebből a vezérlőgombot nyomkodja a flex irányításához.

9. Rajzoljon az alábbi programrészletnek megfelelő UML2 osztálydiagramot! Csak olyan jellemzőket ábrázoljon, amely a kódrészletből kiolvasható! (6 pont)

```
public interface Pelda {.....}
public class Hallgato {
    private Jegyzet j;
    public int vizsgaz(Pelda p) {
        int n = j.puskaz(p); ...}
}
```

10. Egészítse ki az alábbi UML 2 állapotdiagramot (state chart) a következő leírás alapján! (8 pont)

Egy objektum **Korhely** állapotában 4 alállapot található (**részeg, fickós, macsó, zombi**). A **korhely** állapotba egy belépési ponton (entry point) léphetünk be (**pia**), és két kilépési ponton (exit point) hagyhatjuk el (**detox, gödör**). Ha nem pia-n lépünk be, akkor abba az állapotba kerülünk, amelyikben utoljára voltunk. Ha nem volt ilyen, akkor a fickósba. A pia-ból a részegbe kerülünk. Ha részeg, folyton böfög. Ilyenkor, ha tévét néz, és box megy, akkor macsó lesz, ha az éden hotel megy, akkor meg fickós. Fickósan mindig kacsint. Fickósból részeg jég hatására, macsóból részeg pofon hatására lesz. Ha macsó és iszik, akkor fickós lesz. Bármikor, amikor abbamarad a fickósság, akkor kakukkol. Ha fickós és iszik, akkor eldől és zombi lesz. Ha fickós és pofont kap, akkor a gödör kilépési ponthoz jut. Zombi állapotból kilépve sóhajt. Zombiból fény hatására a detox-ba kerül.

Eredmények értékelése:

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