

VIZSGA FELADATSOR

SZOFTVERTECHNOLÓGIA

c. tárgyból
2013. január 8.

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. Jelölje, hogy az oszlopokban szereplő Java nyelvi elemek milyen módosítókkal rendelkezhetnek! (7 pont)

	konstruktor	statikus metódus	példány metódus	példány attribútum
abstract	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
final	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
protected	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
static	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
synchronized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
transient	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
volatile	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2. Mi a Rational Unified Process (RUP) életciklus modelljének utolsó fázisa? Milyen tevékenységek tartoznak ebbe a fázisba? (4 pont)

utolsó fázis: _____

tevékenységek: _____

3. Jellemezzünk egy stringet az alábbi műveletekkel!

ADD(s, x) az s string végére rakja az x karaktert.

LGTH(s) az s string karaktereinek számát adja.

NEW() új (üres) stringet hoz létre.

OCN(s, x) az s stringben található x karakterek előfordulásának számát adja.

TAIL(s) az s string legrégebbi karakterének levágása után maradó stringet adja.

Az alábbi kifejezésekhez adja meg, hogy azok algebrai axiómák BAL oldalán állhatnak-e vagy sem! (5 pont)

Igen	Nem		Igen	Nem	
<input type="checkbox"/>	<input type="checkbox"/>	OCN(TAIL(s), x)	<input type="checkbox"/>	<input type="checkbox"/>	ADD(NEW(), x)
<input type="checkbox"/>	<input type="checkbox"/>	LGTH(OCN(s, x))	<input type="checkbox"/>	<input type="checkbox"/>	OCN(ADD(s, y), x)
<input type="checkbox"/>	<input type="checkbox"/>	OCN(ADD(s, x), x)	<input type="checkbox"/>	<input type="checkbox"/>	OCN(NEW(), 0)
<input type="checkbox"/>	<input type="checkbox"/>	TAIL(OCN(s, x))	<input type="checkbox"/>	<input type="checkbox"/>	LGTH(TAIL(s))
<input type="checkbox"/>	<input type="checkbox"/>	LGTH(ADD(NEW(), x))	<input type="checkbox"/>	<input type="checkbox"/>	ADD(s, LGTH(s))

4. Egy UML2 modelben legyen egy Student osztályunk. Daniel a Student osztály valós idejű példányának UML2-beli modellje.

Kinek a példánya Daniel? (2 pont):

Kinek a példánya Student? (1 pont):

Az alábbi (nem korrekt!) részletet kiegészítve javítsa az ábrát és jelölje be a Student és Daniel közötti kapcsolatot! (2 pont)

Daniel

Student

5. Az alábbi táblázatban a fontosabb szoftver architektúra típusokat adtuk meg, minden típushoz egy indexet rendelve. A jobboldali táblázatban szoftver architektúrákkal kapcsolatos fogalmak állnak. Adja meg, hogy az egyes fogalmak melyik architektúra típushoz kapcsolhatóak leginkább! (A fogalom előtti mezőbe írja be a típus indexét!) Ha a fogalom egyik típushoz sem társítható egyértelműen, akkor a mezőbe tegyen X-et! (3 pont)

1	esemény alapú implicit hívást használó (event-based implicit invocation)
2	csővezeték+szűrők (pipes and filters)
3	tábla (blackboard)
4	interpreter
5	szolgáltatás orientált architektúra (SOA)

	XML
	szórás (broadcast)
	batch (kötegelés)
	adatfolyam (data flow)
	előfizet (subscribe)
	grafikus interfész (GUI)

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

6. Készítsen Java metódust, amelyik két, X típust tartalmazó halmazt kap paraméterül, és visszaadja a két halmaz uniójának elemszámát. Az eredeti halmazok nem változhatnak meg, az elemekhez egyedileg nem szabad hozzáférni! Ahol lehet, használja a collection framework osztályait és interfészeit! Nem használhat default konstruktort! Törekedjen minél általánosabb megoldásra! (4 pont)

7. A Java kódrészletek alapján rajzoljon UML 2 osztálydiagramot! (8 pont)

```
public interface X {
    void bar();
}
public class B {
    C c;
    public void set(C x) {
        c = x;
        c.xxx();
    }
    public C get() {
        c.hello().bar();
        return c;
    }
}
public class D implements X {
    public void bar() {}
}
```

```
public class C {
    public void xxx() {
    }
    public X hello() {
        xxx();
        return new D();
    }
}
public class A {
    static protected double qux(double d){
        return 2*d;
    }
    public void foo(B b) {
        C c = b.get();
        c.xxx();
    }
}
```

