

Név: ..... Neptun kód: .....

## Szoftvertchnológia és -technikák vizsgadolgozat - Minta

2019.12.31

Ponthatárok	
0-49	Elégtelen (1)
50-59	Elégséges (2)
60-69	Közepes (3)
70-84	Jó (4)
85-	Jeles(5)

Vizsga Pontok		
kérdés	max	pont
1	5	
2	5	

3	28	
4	12	
5	15	
6	10	
7	5	

Pontok	
Hozott	20 /
Vizsga	80 /
Σ	100/

Jegy:

⚠ A vizsga két szakaszból áll. Az első szakaszban lévő két feladat a Beugró, itt mind a két feladatból legalább 2-2, de összesen legalább 5 pontot el kell érni, különben a teljes vizsga értékelése elégtelen. A Beugrót erre a lapra kell írni, 15perc van rá! A Beugró feladatainak pontszáma is beszámít a végső jegybe, így az elérhető maximum pontszám 80 pont. A feleletválasztós kérdéseknél, ha külön nincs említve, akkor pontosan **egy** helyes válasz van.

1. Rajzolj le a szabványos jelölést használva (UML) a következő feladatokat a kihagyott helyre! (5p)

a) Osztálydiagram: A „Client” osztály megvalósítja az „IClient” interfészt

b) Osztálydiagram: A „Könyv” osztály és a „Lap” osztály közti kapcsolat (max 300 lap lehet egy könyvben)

c) Osztálydiagram: A „Szerver” osztály IP címét csak a leszármazott osztályok (ill. saját maga) állíthatják be

d) Állapotgép: Az „Egy” állapotból „Kettő” állapotba jutunk az „átlép” esemény hatására

e) Aktivitásdiagram: A folyamat adott szála véget ér (de a teljes folyamat nem)

2. Tervezési minták/architektúrák (5p)

a) Melyek a Singleton tervezési minta megvalósításának kellékei? Jelölje meg a helyes válaszokat! Rossz válasz megadása pontlevonással jár. (3p)

- |  |  |   |
|--|--|---|
| <input type="checkbox"/> Globális változó    | <input type="checkbox"/> Virtuális metódus                       | <input type="checkbox"/> Absztrakt metódus  |
| <input type="checkbox"/> Statikus tagváltozó | <input type="checkbox"/> Statikus metódus vagy statikus property | <input type="checkbox"/> Védett destruktork |
| <input type="checkbox"/> Származtatás        | <input type="checkbox"/> Védett konstruktor                      | <input type="checkbox"/> Interfész          |

b) Adja meg a Separation of Concerns (SoC, vonatkozások/felelősségi körök különválasztása) elv két előnyét (2p)

1. ....

2. ....

**Szoftvertchnológia és -technikák vizsgadolgozat****2019.12.31.**

3. Készítsen az alábbi specifikációhoz osztálydiagramot! Alkalmazza a tanult tervezési mintákat, ahol szükséges! A szövegben kiemelt fogalmaknak mindenképpen vegyen fel saját **osztályt** vagy *interfészt*! Döntéseit, ahol szükséges, rövid szöveges indoklással egészítse ki! Törekedjen rá, hogy a modell minél kifejezőbb legyen! A rendszer dinamikus viselkedését modellezze szekvenciadiagram segítségével a lenti leírás szerint! Ha a feladatot nem tudja megoldani, részpontért ismertetheti az osztálydiagram elemeit egy másik példán, vagy UML diagramok helyett adhat C# vagy Java kódvázatot! (28p)

Robotok gyártását modellezzük. A **robotok** különböző **komponensekből** állnak. A komponensek lehetnek **összetett komponensek** vagy **alap alkatrészek**. Az összetett komponensek állhatnak további (alap, vagy összetett) komponensekből, maximum 10 darabból. Az alap alkatrészek vonatkozásában az alkatrész azonosítószámát, anyagát és tömegét (lebegőpontos szám) kell nyilvántartani: ezeket létrehozásuk után csak lekérdezni lehet kívülről, megváltoztatni nem. Az összetett komponensek esetében csak az azonosítószámot tartjuk nyilván, de a tömeg is lekérdezhető: ilyenkor mindig újra ki kell kiszámolni az értéket. Az azonosító és a tömeg egységes formában legyen lekérdezhető az összetett komponensekre és az alap alkatrészekre. Az alkatrészek három különböző típusú anyagból készülhetnek: alumínium, réz és műanyag. Az összetett komponensek támogatják egy komponens (mely lehet összetett is) beszerelését és kiszerezését.

A robot gyárban *gyártási ellenőrök* dolgoznak, akik folyamatosan figyelik az összetett komponenseken történt kritikus változásokat, amit a beszerelés és eltávolítás folyamat okozhat. Jelenleg két típusú gyártási ellenőr van a gyárban, aki értesíteni tud a változásokról: **mérlegelő**, aki a tartalmazott komponensek tömegét ellenőrzi (minden változásnál) és **minőségbiztosító**, aki pedig a tartalmazott komponensek azonosítóját ellenőrzi. Ügyeljen arra, hogy minél könnyebb legyen új típusú gyártási ellenőröket bevezetni!

Jellemezze a dinamikus működést egy robotkaron (összetett komponens) keresztül: egy új mérlegelő feliratkozik a komponens értesítéseire, majd a robotkarba egy új szervomotort (alapkomponeus) szerelünk be.

4. Készítsen használati eset (Use Case) diagramot az alábbi specifikáció alapján! (12p)

Egy kisboltban zajló vásárlás folyamatát szeretnénk modellezni. A boltba betérő vásárló szétnézhet a boltban, ekkor az eladó segíthet neki, de csak akkor, hogyha épp nem elfoglalt. Ha a vásárló vásárolni akar, akkor kiválasztja a terméket, leveszi a polcról, és a kosarába teszi. A vásárlónak lehetősége van fizetni is az eladónál, ekkor eldönti, hogy mivel szeretne fizetni (hitelkártya, vagy készpénz). A fizetés részletes folyamata nem lényeges a rendszerünk szempontjából. Amikor a boltban nem tartózkodik egy vásárló sem, az eladó szünetet tart, egészen addig, amíg egy újabb vásárló nem érkezik a boltba. A vásárlók között megkülönböztetünk törzsvásárlókat is, ők általában tudják, hogy miért jönnek, de néha ők is szétnéznek a boltban. Törzsvásárlóként kérhető házhozszállítás is, ami egyébként nem.

5. Jellemezze a "Proxy" tervezési mintát! Mire ad megoldást a "Proxy" tervezési minta? Mutassa be általánosságában vagy a minta egy alkalmazásán keresztül (elég az egyik) a minta működését! Ezen belül rajzolja fel a minta osztálydiagramját, valamint röviden adja meg a mintában szereplő osztályok szerepét! Az osztálydiagramon a minta működésének szempontjából kritikus metódusok esetében pseudokódot is adjon meg!

Megvalósításra konkrét forráskódot nem kell írnia, ugyanakkor részpontszámért UML diagram helyett C# vagy Java kód alapján is ismertetheti a mintát. (15p)

6. Foglalja össze a DevOps céljait, az általa megoldott problémák lényegét 8-10 mondatban! Mit jelent a

7. Jelölje a jellemzően összetartozó fogalmakat a két csoport elemei közt! Lehetséges, hogy van elem, amihez nem tartozik pár és az is lehet, hogy van olyan, amihez több pár is tartozik. A rosszul jelölt kapcsolat pontlevonással jár. Döntéseit indokolhatja röviden, ha szükségesnek érzi! (5p)

- |                      |                                |
|----------------------|--------------------------------|
| a. Unit teszt        | 1. Alpha-Beta teszt            |
| b. Modul teszt       | 2. Egy-egy metódus tesztelése  |
| c. Integrációs teszt | 3. Külső cég is végezheti      |
| d. Rendszerteszt     | 4. Memóriaszivárgás tesztelése |
| e. Átvételi teszt    | 5. Kész terméket teszteli      |
| f. Regressziós teszt |                                |