

Szoftvertchnológia és -technikák

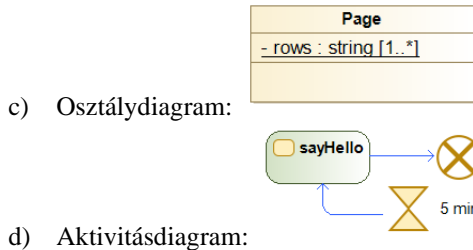
2020. 11. 06. Zárthelyi – Minta ZH

Név:

Neptun kód:

Feladat	Pontszám
1	
2	
3	
4	
5	
6	
7	
8	
Σ	
Jegy	

- Válaszoljon a következő kérdésekre 2-3 (!) mondatban! (8p)
 - A SOLID elvek közt mit jelent az „L” betű?
 - Mi az UML Profile? Nevezzen meg egyet!
- Értékelje az alábbi állításokat 1-2 mondatban! (6p)
 - Az osztálydiagramon megadhatóak az attribútumok, azok kezdőértékei, a metódusok és azok implementációja is.
 - Szekvenciadiagram esetén az aszinkron függvényhívásoknál a visszatérési pontot az <<async>> sztereotípiával jelöljük.
 - A használati esetek közt két sztereotípiát (<<include>> és <<exclude>>) használhatunk. Előbbi kiterjesztést (benne foglalást), míg utóbbi kizárást ad meg
- Adja meg az alábbiakban található fogalmak jelölését (elég egy diagram részlet, nem kell teljes modellt adnia), vagy fejtse ki a megadott jelölés jelentését pontosan 1-2 mondatban! (12p)
 - Szekvenciadiagram: az egyik objektum elkér egy token a másiktól, amit az meg is ad neki
 - Állapotgép: ha ZH-t írok és puskázáson kapnak, véget ér a tevékenységem



- Adja meg az alábbi specifikációhoz készített osztálydiagramot! Könnyítésképpen néhány osztály/interfészt megadtunk, amiket fel kell használnia. Döntéseit, ahol szükséges, rövid szöveges indoklással egészítse ki! Törekedjen rá, hogy a modell minél kifejezőbb legyen! (20p)

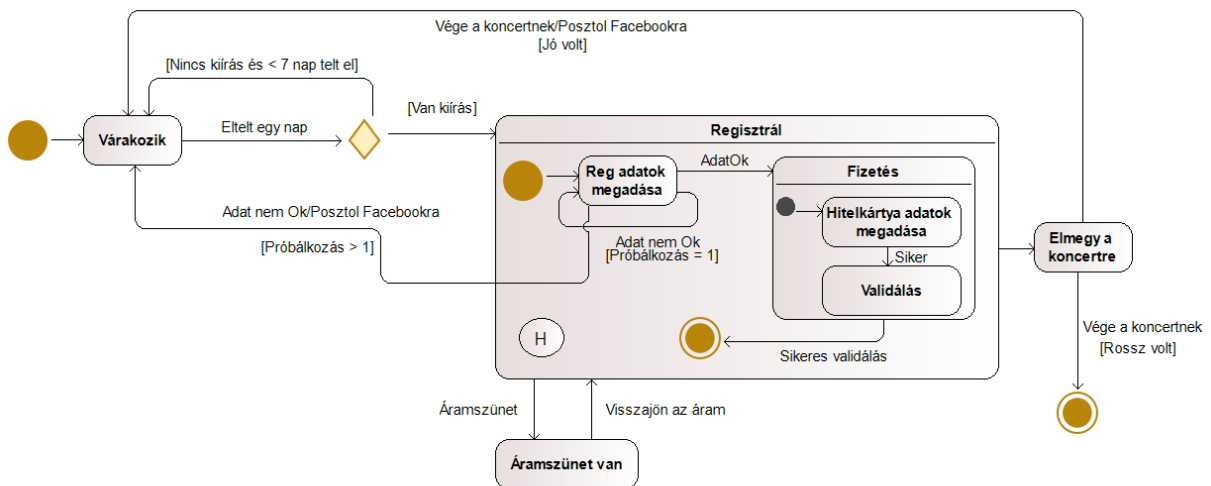
A Zöldlomb kertészetben növények termesztésével foglalkoznak. Minden növénynek van fajtája, amely az adott fajhoz tartozó tulajdonságokat egyértelműen meghatározza. A fajtáknak a latin nevét és a napokban mért tenyészidejét tartjuk nyilván. Minden fajtának lehetnek továbbá káros szomszédfajtái, amelyek rossz hatással vannak az adott fajtájú növények fejlődésére. A növények termesztésében fontos szempont a növény környezetének hőmérséklete. Ha egy növény hőmérséklete túl magas lesz, akkor az valamilyen reakciót vált ki belőle, ami a növény belső folyamata. A kiváltott reakciók kiszámíthatatlanok, minden növény esetleg máshogy reagálhat. A kertészetben jelenleg kétféle konkrét növényt termesztenek: földiepret és fokhagymát. A növények között megkülönböztetjük a zöldségeket, amelyeket elő kell készíteni szállításra a helyi zöldségpiacra. Minden zöldséget máshogy kell előkészíteni. A növényeket üvegházban termesztik. Egy üvegház növényeket tartalmaz és tároljuk róla a belmagasságát. Az üvegházakhoz növények egész sorát hozzá lehet adni. A hozzáadás néha sikertelen is lehet, ha nincs elég hely az üvegházban. Minden üvegházban kertészek dolgoznak. Egy kertész ismeri az üvegházakat, ahol tevékenykedik és minden üvegházban nyilvántartják az összes ott dolgozó kertészt. A növényeket persze öntözni is kell, ez a kertészek feladata. Egy növény öntözése kétféle eredménnyel zárulhat: a növény vagy hidratált lesz, vagy elfonnyad, mert túl sok vizet kapott.

Felhasználható osztályok: Kertész, Üvegház, Eper, Fokhagyma

- 5) Készítsen szekvenciadiagramot az alábbi kódrészlet alapján! A szekvenciadiagram a **OnlineLottery** objektum **iWannaBeABillionaire()** metódusának meghívásával kezdődjön! Ügyeljen a helyes jelölések használatára! (12p)

<pre> public class OnlineLottery { private Player player; private Ticket ticket; private GamblingSystem sys; public OnlineLottery(Player p, Ticket t, GamblingSystem s) { player = p; ticket = t; sys = s; } public void iWannaBeABillionaire() { if (player.getAge() < 18) { showAgeLimitMessage(); } else { showAd(); int selectedNumbers = 0; while(selectedNumbers<5) { int n = player.selectNumber(); ticket.addNumber(n); selectedNumbers++; } player.pay(ticket); sys.validate(ticket); } } public void showAgeLimitMessage() {} public void showErrorMessage() {} public void showAd() {} } </pre>	<pre> public class Player { private String name; private int balance; private int age; public Player(String name, int balance, int age) {} public int getAge() { return age; } public int selectNumber() {} public void pay(Ticket t) {} } public class Ticket { List<Integer> numbers; public Ticket() { numbers = new ArrayList<Integer>(); } public void addNumber(int num) {} public void deleteNumber(int i) {} } public class GamblingSystem { public void validate(Ticket t) { checkNums(t); register(t); } private void checkNums(Ticket t) {} private void register(Ticket t) {} private void reject(Ticket t) {} } </pre>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

- 6) Indokolja az alábbi állítások igazságtartalmát az alábbi állapotgép alapján! Csak az indoklásért jár pont, az állítás igazságtartalmának megállapításáért (igaz / hamis) nem. Az állapotgép egy koncertre járó ember viselkedését írja le, az ember állapotaira fókuszálva. (12p)



- A következő eseménysorozat hatására az ember eljuthat egy koncertre: „Eltelt egy nap” → „AdatOk” → „Áramszünet” → „Visszajön az áram” → „Siker”.
- A koncertre járó ember előbb-utóbb mindenképp eljut egy koncertre.
- Áramszünet után, az áram visszajövelekor a regisztrációs adatokat mindenképp ismét meg kell adni.
- Áramszünet bármikor bekövetkezhet a regisztrációs folyamat végzése során.

- 7) Jellemezze röviden (4-5 mondatban) a “Composite” tervezési mintát! Mire ad megoldást a “Composite” 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 adja meg a mintában szereplő osztályok szerepét! Ahol releváns, pszeudokódot is adjon meg! Megvalósításra konkrét forráskódot nem kell írnia (ugyanakkor részpontoszáért UML diagram helyett C# vagy Java kód alapján is ismertetheti a mintát). (15p)
- 8) Egy alkalmazásban az ön feladata felhasználói felület fájlmegnyitás (Fájl/Megnyitás menü) műveletének kezelése. Ez a következő lépésekből áll:

- (1) <Fájllelési útvonal bekérése a felhasználtól>
- (2) Ha az előző lépésben visszaadott útvonal üres string, akkor a fájlmegnyitás művelet megszakítása (a további lépéseket nem hajtjuk végre)
- (3) <Fájl tartalmának elővalidációja>, és kivétel dobása, ha ez hamis (false) visszatéréssel érvénytelen fájl jelez
- (4) <Fájl tartalmának beolvasása>

Adja meg a feladat C# vagy Java nyelvű megoldását! Minden „< >” jelek közötti lépést egy művelet (függvény) reprezentáljon kódban, ezen műveletek törzsében a tényleges logikát ne valósítsa meg (hagyja üresen őket)! A megoldás során a Template Method (Sablonmetódus) tervezési mintát alkalmazva gondoskodjon a következőkről:

- (1) A kódban minél kevesebb kódduplikáció legyen!
- (2) A megvalósítás legyen könnyen kiterjeszthető a <fájl tartalmának elővalidációja> és a <fájl tartalmának beolvasása> lépések vonatkozásában, illetve ez utóbbihoz nem is létezik alapértelmezett megvalósítás. Mutasson is példát a lépések kiterjesztésére!
- (3) Csak azon osztályokat vezesse be, melyek a feladat megoldásához szorosan kapcsolódnak!

A menü megvalósításával, illetve kódjának a menükiválasztás eseményre való bekötésével nem kell foglalkoznia, de azt adja meg, hogy a menü kiválasztásakor az ön által megvalósított osztályok közül melyiket kell példányosítani és az osztály műveletét kell meghívni! Rövid szöveges magyarázattal ismertesse, hogy a megadott megoldás milyen formában biztosítja az elvárt kiterjesztheséget. (15p)