

3A Házi feladat

Felügyeletre tervezés

FIGYELEM: A házi feladat megoldása előtt olvassa el a tárgy weblapján lévő HF tudnivalókat! A házi feladat leadása előtt nézze végig a HF tudnivalóknál szereplő ellenőrző listát!

Kerettörténet

Cégünk marketingvarázslói új stratégiát dolgoztak ki a vállalat bevételeinek növelésére. Ennek értelmében a cég mostantól nem légvédelmi rendszerek bizonyos szoftverkomponenseinek fejlesztésével fog foglalkozni, hanem jópofa, közösségi portálokon és okostelefonon elérhető játékprogramok készítésével. Az elképzelt üzleti modell manapság igen divatos: a játékok ingyenesen játszhatóak, azonban a játékbeli extráért, segítségért fizetni kell. Ahhoz, hogy az üzleti részleg ezeket pontosan be tudja árazni, viszont szükség van arra, hogy a játékok használatával kapcsolatos információkat, statisztikákat gyűjteni tudjuk (pl. hányan játszanak egyszerre, hányan próbálják meg az extra funkciókat igénybe venni, hányan nem tudják valami hiba miatt befejezni a játékot stb.). Ehhez kell nekünk a megfelelő technológiai háttérrel megteremtünk.

Az új koncepció demonstrálására az egyik kollégánk elkészítette egy egyszerű, a jól ismert akasztófajátékot megvalósító program prototípusát. A játék „üzleti logika” része működik, a csillogó-villogó mobilos-webes klienssel egyelőre még nem foglalkozunk, jelenleg a program felügyelhetőségét kell megoldanunk.

Első feladatunk a következő távfelügyeleti lekérdezések és beavatkozások megvalósítása:

- Új rejtvények felvétele, rejtvények módosítása, törlése, rendezése.
- Kapcsolatok számának és különböző adatainak (IP-cím, port) lekérdezése, kapcsolat bontása.
- A program leállítása.

A távfelügyelet megoldására a legalkalmasabb megközelítés az alkalmazás **JMX instrumentációval** történő ellátása.

A program használata

A program az alábbi paranccsal indítható:

```
java -jar Hangman.jar <port>
```

A szerverhez a játékosok (a platformfüggetlenség jegyében) *telnet* segítségével csatlakozhatnak¹:

```
telnet <IP-address> <port>
```

¹ Windows 7-ben már nem települ a telnet kliens, helyette pl. a putty használható.

1. feladat: Ismerkedés az alkalmazással

Hogy jobban megértsük az alkalmazás működését, a forráskód tanulmányozásával készítsen egy UML modellt (tipikusan komponens vagy osztálydiagramokkal), ami ábrázolja a rendszer főbb részeit és azok kapcsolatait. A modell tartalmazza, hogy melyik komponens milyen adatot tárol, hogyan kommunikálnak egymással a komponenseink, tipikus esetben melyikből hány darab fut stb. A modell elemeit, és így az alkalmazás felépítését és működését, egy rövid szöveges leírásban is ismertesse.

Figyelem: nem egy reverse engineering eszközzel előállított részletes osztálydiagramot kérünk, hanem egy saját modellt, ami csak az alkalmazás fontosabb részeit ábrázolja!

2. feladat: Felügyeleti modell elkészítése

Az MBean-alapú instrumentációt előkészítendő és dokumentálandó, UML segítségével adja meg az alkalmazás felügyeleti modelljét! Jelen kontextusban a felügyeleti modell minimálisan egy statikus struktúra diagram, mely megadja a felügyeletet lehetővé tevő MBean(ek) menedzsment-interfészét és az azokat megvalósító osztályok kapcsolatát az alkalmazáslogikáért felelős osztályokkal.

- A felügyeleti modellezés során figyeljen arra, hogy a menedzsment-tevékenységeket alapvetően ne a funkcionalitást megvalósító osztályok végezzék! (Lásd: Rendszermonitorozás előadás.)
- Az egyes, külső felügyelet számára elérhetővé tett attribútumokat/metódusokat dokumentálja úgy, hogy abból a megvalósítást nem ismerő szereplők számára is egyértelmű legyen azok jelentése!

Röviden ismertesse, hogy a felügyelet megvalósításához milyen változtatások szükségesek a már adott osztályokon, illetve alkalmazás-logikán!

3. feladat: Felügyeleti működés megvalósítása JMX segítségével

Valósítsa meg a fent megtervezett JMX alapú távfelügyeleti műveletek közül **ötöt**, és dokumentálja a megvalósításhoz kapcsolódó főbb tervezési döntéseket! Mutassa be a megvalósítás távolról vezérlését és az ahhoz szükséges indítási paraméterezést, valamint környezet-konfigurációt!

További követelmények

- Az alkalmazás publikus elemeihez kötelező Javadoc típusú kommenteket készíteni.
- Felhívjuk a figyelmét, hogy a dokumentáció mellett a forráskód, JAR állományként a lefordított kód és egy indítószkript is leadandó.
- A megoldás során lehet, hogy módosítani kell az eredeti forráskódot. A forrásfájlok új verzióján kívül a leadott csomagba rakjon be *diff* fájlokat is (unified diff formátumban²), melyek a módosításokat jelzik.

² Wikipedia. Diff, Unified format, http://en.wikipedia.org/wiki/Diff#Unified_format