

3C 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 hátteret 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.

Hogy a felhasználók viselkedését jobban nyomon tudjuk követni, és az esetleges hibák okát könnyebb legyen kideríteni, az első feladatunk az alkalmazás kiegészítése naplózással.

A probléma megoldására a legalkalmasabb megközelítés az alkalmazás kiegészítése úgy, hogy a **log4j** keretrendszer segítségével naplózzon.

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. feladat: Ismerkedés az alkalmazással

Hogy jobban megértsük az alkalmazás működését, a forráskód tanulmányozásával készítsen egy UML modellt (tipikusan osztálydiagramokkal), ami ábrázolja a rendszer

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

főbb részeit és azok kapcsolatait. A modell elemeit, és így az alkalmazás felépítését és működését, egy rövid szöveges leírásban is ismertesse.

Készítse el ezen kívül a **GameSession** osztály viselkedését leíró dinamikus modellt is *UML állapotgépek* formában.

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

Megfelelő érveléssel alátámasztva sorolja fel, hogy a kiszolgáló komponensnek milyen, a működésével kapcsolatos eseményeket kellene rendszerfelügyeleti célokra elérhetővé tennie! Az eseményeket adja meg egy táblázatban is, legalább a következő adatokkal:

- esemény azonosítója (integer),
- esemény súlyossága,
- esemény szövege,
- az esemény definíciója (annyira formálisan és részletesen, hogy az további pontosítás nélkül implementálható legyen).

Legalább **nyolc** esemény definiálását várjuk el, ezek között kell lennie Error vagy Warning, Information és Verbose súlyosságú eseménynek is.

3. feladat: Naplózás megvalósítása

Log4j instrumentáció hozzáadásával és a szükséges egyéb módosításokkal érje el, hogy a kiszolgáló a fent definiált eseményeket naplózza! Felhívjuk rá a figyelmét, hogy a kiadott kódban a hibakezelés szándékoltan részleges és elnagyolt, annak rendbetétele a feladat részét képezi.

A feladat megoldásának része az a mérnöki érvelés, hogy hány különböző helyre és milyen logika szerint naplóz az alkalmazás.

További követelmények

- A tesztelés tartalmazza az összes esemény kiváltásának módját és azok naplózásának bemutatását.
- 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