

## 3B 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.

A maximális profit elérése érdekében az első feladatunk az, hogy a programnak alkalmasnak kell lennie a felhasználók viselkedésének nyomon követésére, ezért felmerült az igény az alkalmazás teljesítményszámlálókkal történő instrumentálására.

A teljesítményszámlálók implementálására 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 csatlakozhatnak<sup>1</sup>:

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

### **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

---

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

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**

Megfelelő érveléssel alátámasztva sorolja fel, hogy a kiszolgálónak milyen, a működésével kapcsolatos kvantitatív jellemzőket kellene rendszerfelügyeleti célokra elérhetővé tennie (pl. uptime)!

A metrikák definícióját valami jól áttekinthető formában adja meg (pl. táblázat), mely legalább a következő tulajdonságokat tartalmazza:

- metrika neve,
- metrika leírása,
- metrika számításának módja.

Ügyeljen arra, hogy a metrikák definícióját annyira formálisan és részletesen adja meg, hogy azok további pontosítás nélkül implementálhatóak legyenek (ez nem azt jelenti, hogy konkrét változóneveket kell megadni, csak annyit, hogy a számolásuk pontos mikéntjét leírjuk).

Legalább **öt metrika** definiálása kötelező. A metrikák között lennie kell sima számláló típusú, átlagot és rátát nyilvántartó metrikának is.

## **3. feladat: Teljesítményszámlálók megvalósítása JMX segítségével**

Valósítsa meg a fenti metrikák nyilvántartását az alkalmazásban, és tegye lehetővé azok **JMX** segítségével történő lekérdezését. A megvalósítás során figyeljen arra, hogy a menedzsment-tevékenységeket alapvetően ne a funkcionalitást megvalósító osztályok végezzék!

A dokumentációban a megvalósítás leírásán kívül mutassa be a teljesítményszámlálók lekérdezésének működését, és ismertesse a szükséges környezeti beállításokat.

## **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 formátumban<sup>2</sup>), melyek a módosításokat jelzik.

---

<sup>2</sup> Wikipedia. Diff, Unified format, [http://en.wikipedia.org/wiki/Diff#Unified\\_format](http://en.wikipedia.org/wiki/Diff#Unified_format)