

Segédlet a Rendszermodellezés (VIMIAA00) szorgalmi házi feladathoz

Kritikus Rendszerek Kutatócsoport

2022

Tartalomjegyzék

1. A szükséges komponensek telepítése	1	3. Tesztek írása és végrehajtása	3
1.1. Cucumber	1	3.1. Tesztek írása	3
1.2. EclEmma	2	3.2. Tesztek végrehajtása	3
		4. Feladatkiadás és feladatbeadás	4
2. A szükséges infrastruktúra beállítása	2	4.1. Feladat kiadása	4
2.1. Projekt importálása	2	4.2. Feladat beadása	4
2.2. Build path beállítása	2	4.3. Feladat védeése	4

Bevezetés

Jelen segédanyag a BME VIK elsőéves informatikus hallgatói számára készült a *Méréstechnika és Információs Rendszerek Tanszéken* Lucz Soma és Farkas Rebeka munkájának felhasználásával, és a Rendszermodellezés (VIMIAA00) című tárgy szorgalmi házi feladatainak elkészítésében segít.

Figyelem! A szöveg a *Yakindu* 3.5.2-es verziójával, a *Cucumber* 0.0.23-as, és az *EclEmma* 3.1.3-as verziójával van összhangban. Nyomatékosan kérjük, hogy idén (2021-ben) a házi feladat elkészítéséhez is ezeket a verziókat használják, mert más (akár újabb, akár régebbi) verziókkal kompatibilitási probléma léphet fel!

1. A szükséges komponensek telepítése

A szorgalmi feladat elkészítése kapcsolódik a kötelező házi feladathoz, így az ahhoz szükséges környezet megléte elvárt. Ezt a kötelező házi feladathoz készült segédletben mutatjuk be. Jelen fejezetben feltételezzük, hogy ez a telepítés sikeresen lezajlott.

1.1. Cucumber

A tesztelést Gherkin nyelven kell majd elvégezni, ehhez nyújt felületet a Cucumber. Ennek telepítése a következő módon történik:

- Meg kell nyitni azt az Eclipse telepítést, amiben a házi feladat készült
- *Help > Install new software*
- A *Contact all update sites...* opció **ne** legyen kiválasztva!
- Írjuk be az alábbi repository címet a *Work with* mezőbe: <https://cucumber.github.io/cucumber-eclipse/update-site>

- Válasszuk ki a *Cucumber Eclipse Plugin* opciót
- A telepítés innentől Next, Next, Finish jellegű

A telepítést követően meg fogja kérdezni a telepítő, hogy újraindítsa-e az Eclipse-t, ám erre még nincsen feltétlenül szükség.

Megjegyzés. A telepítés során felteheti a kérdést a telepítő, hogy biztosan telepíteni akarjuk-e, mert nincsen érvényes tanúsítványa a telepítendő komponensnek. Ebben az esetben válasszuk az *Install anyway* lehetőséget.

1.2. EclEmma

A tesztesetek kiértékelésének fontos eszköze a fedettség mérése. Ennek az elvégzésére fogjuk használni az EclEmma eszközt. Ennek telepítése az alábbiak szerint működik:

- Meg kell nyitni azt az Eclipse telepítést, amiben a házi feladat készült
- *Help > Install new software*
- A *Contact all update sites...* opció **ne** legyen kiválasztva!
- Írjuk be az alábbi repository címet a *Work with* mezőbe: <http://update.eclEmma.org/>
- Válasszuk ki a *EclEmma Java Code Coverage* opciót
- A telepítés innentől Next, Next, Finish jellegű

A telepítést követően meg fogja kérdezni a telepítő, hogy újraindítsa-e az Eclipse-t, ezt most **tegyük meg**.

Megjegyzés. A telepítés során felteheti a kérdést a telepítő, hogy biztosan telepíteni akarjuk-e, mert nincsen érvényes tanúsítványa a telepítendő komponensnek. Ebben az esetben válasszuk az *Install anyway* lehetőséget.

2. A szükséges infrastruktúra beállítása

2.1. Projekt importálása

Kezdeként, le kell tölteni a feladatkiírás oldaláról a kiadott Cucumber-es tesztelő projektet. Ez egy .zip fájl lesz. Ezt követően:

- Nyissuk meg azt az Eclipse példányt, amibe az eszközöket telepítettük:
- *File > Import... > Existing Projects into Workspace*
- Válasszuk a *Select archive file* opciót, és keressük meg a letöltött .zip állományt.
- *Finish*

A projektben ezt követően még fordítási hibák lesznek, mert hiányoznak még a saját házi feladat fájlljai. Ezért a következő hibákat olvashatjuk az Eclipse *Problems* nézetében:

- `The import hu.bme.mit.inf.symod.homework.generic cannot be resolved`
- `TesterAPI cannot be resolved to a type`

2.2. Build path beállítása

A hibák megszüntetéséhez be kell állítani az imént importált projekt build path-ját. Ezt a következő képpen kell megtenni:

- Jobb klikk a *hu.bme.mit.inf.symod.homework.cucumber* projektre > *Build Path > Configure Build Path*
- A megjelenő *Java Build Path* ablak beállításainál válasszuk a *Projects* fület *Add...* > Válasszuk ki a házi feladatunkat > *Ok*

Megjegyzés. Ahhoz, hogy a fenti módszer működjön, a workspace-ben meg kell lennie nyitva a házi feladat projektjének.

Figyelem! A Build path beállításánál figyeljünk arra, hogy mind a házi feladat, mind a Cucumber-es projekt ugyanazzal a JDK-val legyen fordítva!

3. Tesztek írása és végrehajtása

3.1. Tesztek írása

Nyissuk meg a fejlesztőeszközben a kiadott projekt alábbi állományát: *src/-hu.bme.mit.inf.symod.homework.cucumber/chessclock.feature*. Ha mindent helyesen csináltunk, egy külön fejlesztőfelület jelenik meg, amelyben teszteseteket írhatunk a sakkóránkhöz. Alább egy egyszerű példa látható, amely végigpróbálja az összes gombot:

```
Scenario: Basic test: Pressing each button.  
Given a fresh chess clock  
When I press the WHITE button  
And I press the BLACK button  
And I press the MOD button  
And I press the START button  
Then NO BEEP should be made
```

A fentihez hasonló kezdeti példák találhatóak a kiadott állományban, amelyek a Gherkin tesztleíró szintaxis megértését segítik. A kiadott példák nem feltétlenül értelmes tesztesetek, és akár ellent is mondhatnak a sakkóra kiadott specifikációjának; a feladat ezek helyett saját teszteseteket készíteni.

A feladatkiírásnak megfelelően készítsük el saját tesztjeinket a *chessclock.feature* állományba. A tesztleírások szerkesztését gördülékenyebbé tehetjük, ha használjuk a CTRL+Space kombinációt, amely kiírja az adott ponton elérhető összes szerkesztési lehetőséget.

A tesztelőeszközhöz további dokumentációt, segítséget találhatunk a honlapján¹.

3.2. Tesztek végrehajtása

Az elkészített tesztjeinket a *CucumberCoverage.launch* konfiguráció alapján indíthatjuk. Ez amellet, hogy lefuttatja a tesztjeinket, megfelelő beállítások mellett képes a tesztesetek hatékonyságát is mérni különböző kódfedettségi metrikák kiszámolásával.

Teszteszek futtatásához:

- Jobb klikk a *CucumberCoverage.launch* állományra
- *Coverage As... > CucumberCoverage*

A tesztesetek lefutása után a *JUnit* nézet tájékoztat minket az egyes tesztesetek eredményéről. Ha egy teszteset sikertelen, akkor vagy a tesztleírásban, vagy a sakkóra modelljében hibát vétettünk, javítsuk!

A Coverage ablakban pedig megfigyelhetjük a programkódunk fedettségi metrikáit. A kiterjedt fedettségi jelentés sok olyan részletet is tartalmaz, amely a feladat szempontjából nem releváns; minket kifejezetten a Yakindu állapotgépet megvalósító generált kód érdekel, pontosabban a **ChessClockStamemachine.java** állomány. Ezen belül az állapotátmeneteket megvalósító `check_`, `react_`, `effect_` ill. az állapotok be- ill. kilépésekor végrehajtott `enterSequence_`, `exitSequence_` kezdetű függvények teljes fedése a cél. Erre szolgál a nézet jobb felső sarkában található lefele mutató háromszög által kiválasztható *Method Coverage* mód, de az alapértelmezett *Instruction Coverage* is használható. Nem feladat lefedni a generált kód egyéb, pl. hibakezelési célú részeit.

A fedettségnérő eszközhöz további dokumentációt, segítséget találhatunk a honlapján².

Megjegyzés. A futtatás során a *Console* nézetben warning-okat fogunk látni, amik érvénytelen reflection műveletre figyelmeztetnek a Cucumber-en belül. Ezeket hagyjuk figyelmen kívül!

¹<https://cucumber.io/cucumber-eclipse/>

²<http://www.eclemma.org/userdoc/index.html>

4. Feladatkiadás és feladatbeadás

A Rendszermodellezés tárgy Moodle oldalán lesz közzétéve a házi feladatokhoz tartozó feladatkiírás illetve a tudnivalók.

4.1. Feladat kiadása

A feladat kiírása megtalálható a tárgyhonlap **Házi feladatok** fülén. A feladat kiírásához tartozik ez a segédlet, továbbá egy `.zip` fájl, ami a kiinduló Cucumber-es projektet tartalmazza.

4.2. Feladat beadása

A házi feladat beadása és a megfelelő fájlok feltöltése a HF portálon keresztül történik. A portálon az *Extra / IMSc feladat (Tesztelés)* pontnál szereplő kérdések megválaszolásával és a tesztelés dokumentálásával teljesíthető a feladat. A beadási határidő a HF portálon a feltöltési határidő.

4.3. Feladat védelme

A feladatot a beadás után meg is kell védeni. Az ezzel kapcsolatos információk a tárgyhonlapon lesznek kihírdetve.