



Budapesti Műszaki és Gazdaságtudományi Egyetem  
Méréstechnika és Információs Rendszerek Tanszék

## **Automatikus tesztelés**

**Szántó Tamás III. évf, (BSc) mérnökinformatikus szakos hallgató**

**Konzulensek: Honfi Dávid doktorandusz, MIT**

**dr. Micskei Zoltán adjunktus, MIT**

**Rendszertervezés szakirány**

**Témalaboratórium összefoglaló**

**2016/17. I. Félév**

A témalaboratórium során a tesztelés témakörével ismerkedtünk meg négy fős csoportban Kiss Dániellel, Németh Zsolttal és Reischl Barnabással. Először a tesztelési lehetőségeket, módszereket tekintettük át majd az ezekhez kapcsolódó automatizálási lehetőségeket. A konzultáció keretei között heti rendszerességgel találkoztunk és értékeltük ki az elvégzett munkánkat majd egyeztettük a következő feladatokat.

A témakör áttekintése érdekében ISTBQ (International Software Testing Qualifications Board) sillabuszát olvastuk el. Következő lépésként fejlesztési környezetet választottunk ketten C#, ketten Java alapokon haladtunk tovább.

JUnit tesztek készítéséhez egy GitHub-on elérhető projektet választottunk. Ez gyakorlatilag matematikai algoritmusok Java nyelvű implementációit tartalmazta. Öt darab algoritmust választottunk ki részletesebb vizsgálatra. A már meglévő tesztek körülbelül teljesen lefedték a kódot, de csak egyszerű eseteket vizsgáltak meg. Ezért saját egységteszteket dolgoztam ki hibák felfedezésére törekedve. Ezekkel meglepően sok hibás működést sikerült észlelnem, jellemző típushibák amiket találtam: elvi hiba, tömb túlindexelés, érték túlcsoportolás, StackOverflowError.

A talált hibák esetén egyértelmű volt, hogy ezek nagyrésze olyan jellegű aminek a megtalálása automatizálható kell hogy legyen belátható bonyolultság mellett. Az EvoSuite és Randoop eszközök tesztgenerálási lehetőségeit próbáltam ki az előzőleg kiválasztott osztályokra. Mindkét eszköz működési elve a tesztelt osztály többszöri futtatása meghatározott bemenetekkel, majd ezek közül egy bizonyos szempontrendszer szerint a különböző lefutást eredményező megtartása. Az EvoSuite kicsit komplexebb teszteket hoz létre, mivel a bemeneteket keresés-alapon választja, célja a teljes tesztlefedettség elérése. A Randoop bemeneteit okosan random választja így lényegesen több de egyszerűbb tesztet generál, ezek például alkalmasak lehetnek regressziós tesztelésre.

Az egységtesztelés szempontjából fontos a függőségek (pl.: adatbázis, webszerver, külső modul) megfelelő kezelése. A probléma alapja, hogy a tesztelt rész viselkedését a függőségek egy adott állapota mellett szeretnénk vizsgálni, de nem férünk hozzá a függőségekhez. Erre ad egy megoldási lehetőséget a mock. A mockolás témakörében én a Mockito eszközzel foglalkoztam. A lényege hogy a teszt elején a függőséggel való kapcsolatot leíró interfész segítségével egy mock hozható létre. Majd ezen a mock-on beállíthatók az egyes függvények visszatérési értékei, tehát az hogy a függőség milyen állapota legyen látható a tesztelt egység felől. Ez a két lépés után az általános JUnit tesztrészek használhatók.

Függőségek számításba vétele esetén nem volt könnyű dolgunk tesztgenerálás tekintetében. Tapasztalataink alapján a Java nyelven elérhető céleszközök (EvoSuite, Randoop) nem működnek együtt a Mockito eszközzel. Önmagukban pedig nem alkalmasak a függőségek kezelésére. A problémára sikerült megoldást találnom, így stubok írásával lehetőség nyílt tesztek generálására.

Utolsó lépésként a felhasználói felület tesztelésével foglalkoztunk. Ennek kapcsán példának a WordPress keretrendszert választottam. A teszt alanya a kezdésképpen rendelkezésre álló egyszerű blog jellegű oldal volt. A Selenium IDE Firefox bővítményét használtam. A tesztelés során egy kattintás sorozatot kellett rögzíteni, ez HTML elemek és azokhoz rendelt események sorozataként került eltárolásra. Majd az éppen aktív böngészőablakban tudtam futtatni a felvett teszteket. A kattintás sor felvétele után volt lehetőség Java kódot is generálni, ez könnyen olvasható és módosítható szerkezetű volt.

A témalaboratórium során alkalmam volt részletesebben megismerkedni a tesztelés több részével, kipróbálni egy nagyobb terjedelmű témakör feldolgozását, dokumentálását csoportban dolgozva.