Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

Automatikus tesztelés

Kövér Márton III. évf, (BSc) mérnökinformatikus szakos hallgató
Konzulensek: Honfi Dávid, Micskei Zoltán
Témalaboratórium összefoglaló
2017/18. I. félév

Kitűzött feladatom volt megismerni Java tesztelési eszközöket (JUnit,Mockito, Randoop) és módszertanokat (tesztek implementációja manuálisan, tesztek izolációja és tesztgenerálás eszköz segítségével)

Egy nyíltforráskódú projekttel dolgoztam, ami Java nyelven lett megírva. Mint arra a neve is utal (Algorithms), egyszerű problémákra, mint pl. adatszerkezetek, rendezések, matematikai műveletek string műveletek, mutat megoldást. Ezen projekthez voltak írva unit tesztek, ezeket elemeztek és kiegészítettem saját tesztekkel. Mértem kódlefedettséget, ami majdnem 100%-os volt, de még így is sikerült javítanom rajta.

Függőségnek hívjuk a modulok közötti kapcsolatot. Tesztelésnél problémát jelentenek a külső függőségek, ugyanis nem tudjuk biztosan, milyen értékekkel fog dolgozni a tesztelendő modulunk. A környezet modellezésével tudjuk kikerülni ezt a problémát, ilyenkor mock objektumok segítségével megadhatjuk a függvények visszatérési értékét és ellenőrizhetjük az átadott argumentumokat.

Izolációs keretrendszerből a Mockito nevűvel dolgoztam, egy új projekten, ugyanis az első túl egyszerű volt. Egy űrhajó torpedólövését kellett tesztelnem, amihez át kellett alakítani az eredeti projektet, ugyanis privát argumentumként szerepeltek a helyettesítendő objektumok. Az átalakított osztályból leszármazott teszt űrhajóval dolgoztam, sikeresen teszteltem a lövéseket.

Unit tesztek generálását a Randoop nevű programmal csináltam. Egy nyíltforráskódú és Java környezethez van kialakítva. Sok tesztet generál véletlenszerű módszer segítségével. A kiadott leírás segítségével is nehézséget okozott a Randoop futtatása. A generált tesztek nehezen olvashatóak.

A teljes Algorithms projekten dolgozva sikerült hibákat találnom a Randoop segítségével. Többszöri, alapértelmezetten száz másodperces, futtatás eredményét vizsgálva arra jutottam, hogy ugyan véletlen szerű a tesztek készítése, adott idő alatt a kész tesztek hatékonysága keveset változik. különböző tesztgenerálási idővel dolgoztam ezután, így megfigyelhető, hogy az idő növelésével kezdetben a kódlefedettség is nagy mértékben nő, de nagy értékekre egyre elhanyagolhatóbb lesz a javulás.