



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 feladatomból 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. A projekthez már tartoztak unit tesztek, először ezeket elemeztem, majd kiegészítettem a tesztkészletet újabbakkal. Munkám során kódfedettséget is mértem, ahol közel 100%-ra tornáztam fel a tesztkészletek ezen értékét.

Tesztelésnél problémát jelentenek a külső függőségek, ugyanis nem tudjuk biztosan, milyen értékekkel fog dolgozni a tesztelendő modul. A környezet modellezésével lehet kikerülni ezt a problémát, ilyenkor mock objektumok segítségével megadható a függvények visszatérési értéke és ellenőrizhetők az átadott argumentumok.

A félév során a Mockito izolációs keretrendszerrel dolgoztam. Ehhez egy másik projektet választottam. 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 végeztem. Egy nyíltforráskódú és Java környezethez van kialakítva. A félév során szerzett tapasztalataim alapján sok tesztet generál véletlenül alapuló módszer segítségével. A Randoop futtatása a kiadott leírás ellenére is sok nehézségbe ütközött. Tapasztalataim szerint az egyik legnagyobb problémája az eszköznek, hogy a generált tesztek nehezen olvashatóak.

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