

Programok statikus analízise

Önálló laboratórium feladat összefoglalója

Csikós Donát (B6RF5)

Konzulens: Majzik István

BME Méréstechnika és Információs Rendszerek Tanszék

Rendszertervezés ágazat, 2008/2009. II. félév

Az önálló laboratórium tárgy keretében két elkülönülő területtel, a statikus programellenőrző eszközökkel, valamint a *Java Modelling Language* nevű specifikációs nyelvvel ismerkedtem meg.

A félév első felében olyan programellenőrző eszközöket vizsgáltam, melyek képesek a Java nyelvű programok forráskódját, illetve lefordított bytekódját beolvasva valamilyen találgatást tenni arra vonatkozóan, hogy a program készítője hibázott-e annak megírásakor.

A három vizsgált program, melyeket közelebbről is megismertem, a *JLint*, a *FindBugs* és a *PMD* volt. Ezek az eszközök nagy eltérést mutattak a hibakereséshez felhasznált eszközökben, mégis képesek voltak a gyakori hibák kimutatására, melyeket akár forráskód adatfolyam gráfjának, akár a bytekód mintáinak figyelésével találtak meg.

A közelebbi vizsgálatokhoz elkészítettem néhány olyan Java programot, melyekben szándékosan egyszerű, mégis tipikus hibákat helyeztem el annak megfigyelésére, hogy az adott típusú problémát milyen hatékonysággal derítik fel az eszközök. Az eredményből az derült ki, hogy az egyes eszközök csak közelítőleg tudják megjósolni, hogy milyen hibás kódrészletek szerepelnek a programokban, illetve hogy minden egyes ellenőrző eszköz és megoldás eltérő területen teljesít megbízhatóbban.

A félév második felében a JML eszközkészletével ismerkedtem meg. Ez egy modellező nyelv, mely a Java programok forráskódjának kommentjeiben speciális annotációkkal rendelkezik arról, hogy az általuk leírt osztályok metódusainak és tagváltozóinak mi az elvárt, specifikált viselkedése. Ezt a viselkedést egy módosított Java fordító és futtatókörnyezet képes értelmezni, és hibaként jelezni, ha futás közben a megkötések megsértette a program.

Miután áttekintettem a rendszer alapjait, áttanulmányoztam a hozzá tartozó eszközök skáláját, hogy átfogó képet kapjak, milyen alkalmazási területei is vannak. A félév vége felé pedig egy saját programon kezdtem el dolgozni, mely a *jmlunit* nevű tesztelő eszközhöz generál a forráskód alapján automatikusan robusztusság teszteket. Ehhez a program először forrásfájlt beolvasva megvizsgálja az egyes előfeltételeket, melyek alapján összeállítja a szükséges tesztadatokat. Ezután a *jmlunit* futtatásával ellenőrzi, hogy a bemeneti adatokra az állomány futása során sérülnek-e a specifikációban szereplő utófeltételek.

A félév végére a program primitív adattípusú paramétereket váró függvények esetén képes egyszerű szélsőértékekkel feltöltött teszteket generálni és futtatni.

Források:

JLint: <http://artho.com/jlint/>
FindBugs: <http://findbugs.sourceforge.net/>
PMD: <http://pmd.sourceforge.net/>
JML: <http://www.cs.ucf.edu/~leavens/JML/>
JML tools: <http://sourceforge.net/projects/jmlspecs/>