

*Futásidejű kivételek detektálása Java programokban szimbolikus végrehajtás segítségével*

*Kádár István*

*III. évf. programtervező informatikus Bsc.*

*Témavezetők: Dr. Ferenc Rudolf egyetemi adjunktus, Hegedűs Péter doktorandusz*

*SZTE TTIK Szoftverfejlesztés Tanszék*

Nagyméretű, megbízható és robusztus szoftverrendszerek előállítása a szoftverfejlesztés egy nagy kihívást jelentő feladata manapság. A fejlesztési költségek nagyjából 40%-a tesztelési költség, továbbá az üzemeltetés során fellépő hibák megtalálása és kijavítása is jelentős erőforrásokat emészt fel és nagy költségekkel jár. Céloom egy olyan módszer és eszköz kidolgozása, amely a szoftverfejlesztés ezen fázisát támogatja azáltal, hogy a Java programokban képes futásidejű kivételeket felderíteni, valamint olyan veszélyes kódrészletekre rámutatni, amelyek mintegy időzített bombaként viselkednek a szoftver későbbi fejlesztéseinek folyamán. Mindezt anélkül szeretnénk elérni, hogy a programot valós környezetben futtatnunk kellene.

A megvalósítás a szimbolikus végrehajtás módszerének felhasználásával történik. Szimbolikus végrehajtás során a vizsgálandó programot nem konkrét input adatokon futtatjuk, hanem szimbolikus változókként kezeljük a bemeneteket. Amikor a program végrehajtása egy a szimbolikus inputtól függő elágazáshoz ér, mindkét lefutási ág végrehajtódik. Ekképpen a vizsgált szoftver minden lefutási ága elméletben végrehajtásra kerül.

A Java PathFinder (JPF) egy a NASA Ames Research Center Robust Software Engineering részlegén kifejlesztett nyílt forrású modellellenőrző program, valójában egy Java virtuális gép, amely a Java bytecode speciális végrehajtására alkalmas. E program kiegészítője a Symbolic PathFinder, amely a Java bytecode utasításokat úgy interpretálja, hogy a futtatás a szimbolikus végrehajtás elméletének eleget tegyen. Munkámhoz ezeket az eszközöket használom fel.

A dolgozatban kitérek arra, hogy hogyan valósítottam meg a Java programozási nyelv RuntimeException-jeinek detektálását a fent említett virtuális gép és a szimbolikus végrehajtás segítségével, meghatározva a kivételt dobó metódus azon paraméterezését is, amellyel az kivételre fut. Kifejtem a módszert, amellyel a lefutási ágak számát és az állapottér méretét csökkentem. Az elkészített alkalmazást a log4j, az ArgoUML és a jEdit nyílt forrású programokon teszteltem.

Kulcsszavak: Szoftverfejlesztés, szimbolikus végrehajtás, Java, kivételkezelés