



Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

Automatikus izolációs eszköz tesztelése

Reischl Barnabás III. évf, (BSc) mérnökinformatikus szakos hallgató
Konzulens: Honfi Dávid doktorandusz, MIT
Szoftverfejlesztés szakirány
Önálló laboratórium összefoglaló
2016/17. II. félév

Önálló laboratóriumi munkám során egy, a BME MIT tanszékén fejlesztett, tesztelést segítő eszköz fejlesztésébe kapcsolódtam be. Ez az eszköz az AutoIsolator, aminek a célja, hogy két, a szoftvertesztelés területén használt technikát, a dinamikus szimbolikus végrehajtást, és a mocking technikát együtt alkalmazva tesztek generáljon az adott programkódhoz. Ehhez az eszköz először a kódot transzformálva a külső függőségeket megszünteti, és egy kontrollált területre irányítja át a hívásokat.

A félév során elvégzett munkámat három részre lehet osztani, ebből az első kettővel párhuzamosan haladtam:

Először meg kellett ismernem az AutoIsolator működésének az alapját, fel kellett telepítenem a saját számítógépre, ki kellett próbálnom működés közben. Ehhez készítettem egyszerű példaprojektet, amiken futtatva az eszközt sikerült is két hibát találnom benne, az egyiket a nem statikus metódusok, a másikat pedig függőségek importálása kapcsán.

Ezzel párhuzamosan a második feladaton a Microsoft által fejlesztett Roslyn keretrendszer megismerése volt. Ez három konkrét feladatot jelentett, amiket egy egyszerű kódon kellett végrehajtanom: Egy metódusnak a bejárásával kellett összeszámolnom a benne előforduló return utasításokat, egy stringet kellett benne kicserélnem, illetve egy külső metódus láthatóságát kellett lekérdeznem.

E két korábbi feladat elvégzése után már volt egy olyan alapismeretem az AutoIsolatorról, ami alapján konzulensemmel, Honfi Dáviddal ki tudtuk választani a nekem való feladatot.

A feladatom az AutoIsolator által végrehajtott kódtranszformáció tesztelése. Két dolgot lehet itt ellenőrizni. Hogy a transzformáció megtörténik-e, és ha igen, akkor helyesen történik-e. Ahhoz, hogy ezt ellenőrizni lehessen, létre kell hozni egy kódbázist, ami minden nyelvi elemet tartalmaz, és eléggé összetett is ahhoz, hogy az AutoIsolator számára kihívást jelentsen. Mivel egy ilyen kódbázis előállítása manuálisan hatalmas munka, ezért az automatikus kódgenerálás mellett döntöttünk.

Először megnéztem két eszközt, amik hasonló funkciót valósítottak meg, ezek azonban nem voltak alkalmasak a feladatra, így nekem kell megírnom egy ilyen eszközt, amire jelenleg CodeGenerator munkanévvel hivatkozom.

Az eszköz felépítése a következő: vannak benne skeleton osztályok, amik nyelvi egységeket, például osztályokat, metódusokat reprezentálnak. Ezekhez vannak konstruktor osztályok, amelyek objektumait a skeleton osztályok konstruktorai várják paraméterként, és tartalmazzák a megvalósítandó elemek számszerűen mérhető adatait. A skeleton osztályok legfontosabb eleme a kódot generáló metódusuk, amik mind olyan objektumokat adnak vissza, amik beilleszthetőek a fentebbi szinten lévő elembe. Ezen kívül a program tartalmaz még segédosztályokat és enumerációkat típuskezeléshez, kivételkezeléshez, a generált kód paramétereinek a karbantartásához.

A CodeGenerator jelenlegi állapotában egyszerű osztályokat tud generálni, amik alapértelmezett tulajdonságokat, és üres, csak return utasításokat tartalmazó metódusokat tartalmaznak.

További munka lesz az eszköz paraméterezhetőségének és nyelvi eszköztárának bővítése, a kódbázis legenerálása, és a generált kóddal az AutoIsolator transzformációjának a tesztelése.