



## SZAKDOLGOZAT FELADAT

**Lucz Tamás Soma**

szigorló mérnökinformatikus mérnökhallgató részére

### Statikus analízis algoritmusok JavaScript környezetben

A mai komplex szoftverek forráskódja akár több millió sornyi programkódból is állhat. A kódminőség biztosítására számos módszer és eszköz létezik, ezek közül az egyik a statikus analízis. A legtöbb programozási nyelv esetén már fejlett eszközök állnak rendelkezésre a statikus analízis támogatására, ám az egyre népszerűbb JavaScript nyelvre ez kevésbé jellemző.

A JavaScripthez jelenleg elérhető statikus analízis eszközök a legtöbb esetben a kód forrásfájljainak egyesével történő elemzését teszik csak lehetővé. Az ilyen dinamikus nyelvek esetén azonban sok olyan összetett eset van, ami több fájl együttes vizsgálatát igényelheti. Egy, már létező, a tanszéken fejlesztett JavaScript elemző keretrendszer, gráfadatbázisokra építve, lehetővé teszi több fájl együttes vizsgálatát, ugyanakkor, a keretrendszer felhasználhatósága jelenleg korlátozott, mert mindössze néhány elemzési algoritmust tartalmaz.

A hallgató feladata, hogy megvizsgálja a rendelkezésre álló statikus analízis algoritmusokat, és a megvizsgált algoritmusok alapján készítsen egy kényszerekből álló tudáshalmazt az előforduló hibákról, amelyet az egyes algoritmusok megtalálnak a forráskódokban. A tudáshalmaz és a vizsgált algoritmusok alapján a hallgató feladata, hogy szisztematikusan válassza ki a tudáshalmazból a leggyakrabban előforduló hibákat és implementálja az őket megtaláló kényszereket az említett, gráfadatbázisra épülő keretrendszerbe.

A hallgató feladatának a következőkre kell kiterjednie:

- Végezzen irodalomkutatást, mutassa be és hasonlítsa össze a különböző programnyelvekben használatos hibakeresési kényszereket! Csoportosítsa a kényszereket eszközönként és programnyelvenként.
- Készítsen egy tudásbázist a hibakereső kényszerekről és az általuk megtalálható hibákról, válassza ki a leggyakrabban alkalmazott kényszereket!
- Tervezze meg, és implementálja a kiválasztott kényszereket egy már meglévő, inkrementális gráfadatbázisokra épülő keretrendszerhez JavaScript forráskódhoz!
- Végezzen méréseket és értékelje az implementált hibamegtalálási kényszereket szisztematikusan kiválasztott, nyílt forráskódú projektek segítségével!
- Értékelje munkáját és vázolja fel a továbbfejlesztési lehetőségeket!

**Tanszéki konzulens:** Honfi Dávid, doktorandusz

**Külső konzulens:**

Budapest, 2017. 03. 11.

.....  
**Dr. Dabóczi Tamás**  
tanszékvezető