

Szegedi Tudományegyetem
Informatikai Intézet

**Ipari JavaScript elemző kiegészítése TypeScript
támogatással**

**Industrial JavaScript analyzer enhancement with
TypeScript support**

Szakedolgozat

Készítette:

Pozsgai Alex

programtervező informatikus BSc
szakos hallgató

Témavezető:

Dr. Antal Gábor

egyetemi docens

Szeged

2023

Feladatkiírás

Manapság kezd a piacon egyre több kódelemző megjelenni, többek között JavaScript programozási nyelvre is. Jelenlegi projekt is egy JavaScript Analyzer. A cél az, hogy a projekt több legyen, mint a piacon a többi kódelemző, ezért bővíteni kell TypeScript támogatással.

A hallgató feladata ennek a programnak(JSAN) a fejlesztése úgy, hogy tudjon TypeScript fileokat és projekteket is kellő pontossággal elemezni. Ezután pedig a meglévő programot optimalizálni, hogy kevesebb erőforrást vegyen igénybe a futtatása, és hogy gyorsabban fusson le.

A megoldási módszerek a hallgató kreativitására vannak bízva.

Tartalmi összefoglaló

A téma megnevezése:

JavaScript Analyzer kiegészítése TypeScript supporttal, JSAN optimalizálása.

A megadott feladat megfogalmazása:

A feladat során el kell érni azt, hogy a JavaScript Analyzer Tool tudjon TypeScript fileokra lefutni, és azokat nagy pontossággal elemezni. Emellett a JSAN működését optimalizálni.

A megoldási mód:

A megoldás során kell változtatni a JavaScriptSchémán, és a JSAN-ban az AstTransformer.js fileban.

Alkalmazott eszközök, módszerek:

A megoldáshoz Visual Studio Code IDE-t, a JavaScriptSchema szerkesztéséhez Visual paradigm-t használta. A projekt lebuildeléséhez és ellenőrzéséhez Visual Studio 2017 programot használtam.

Elért eredmények:

JSAN képes TypeScript fileokat nagy pontossággal elemezni, nagyobb projektre lényegesen gyorsabban fut le, kevesebb erőforrást igényel, mint előtte.

Kulcsszavak:

JavaScriptAnalyzer, TypeScriptAnalyzer, JavaScript, TypeScript, Optimalizálás, Visual Studio Code, Visual Paradigm, Vpp, C++

Tartalomjegyzék

Feladatkiírás	1
Tartalmi összefoglaló	2
Bevezetés	5
JavaScriptSchema átírása	7
1.1. JavaScriptSchemáról	7
1.2. Átírási okok	7
1.3. Nehézségek, problémák	7
JavaScriptAddon szerkesztése	8
2.1. JavaScriptAddonról	8
2.2. Miben változott	8
JSAN2Lim átírása	9
3.1. JSAN2Limről	9
3.2. Bővítések	9
3.3. Miket nem detektált	9
AST Binder Optimalizálás	10
4.1. Binderről	10
4.2. Lassúság okai	10
4.3. Optimalizálás	10
4.4. Eredmény	10
Regteszt frissítés	11

5.1. Röviden a regtesztről	11
5.2. Tesztektről	11
5.3. Tesztek kibővítése	11
Összefoglaló	12
6.1. Program javulása	12
6.2. Tervek	12
6.2.1. JavaScriptSchema naprakészre hozása	12
6.2.2. JSAN további optimalizálása	12
Nyilatkozat	13
Köszönetnyilvánítás	14

Bevezetés

Szakdolgozatomban a JavaScript Analyzer Tool továbbfejlesztése a cél. A továbbfejlesztés arról szól, hogy a javascript mellett előtérbe jön a typescript is. Ezáltal a typescript fileokat és projekteket is elemezni kell. Ez azt eredményezi, hogy a Javascript Analyzer Toolt gyökerestül át kell írni, annak érdekében, hogy JavaScriptet és TypeScriptet is tudjon egyaránt elemezni.

A JavaScript Analyzer Tool az egy nagyobb projektnek az egyik alprojektje. Ezt a nagyobb projektet Analyzer-JavaScriptnek hívják. Az Analyzer-Javascriptnek több alprojektje is van, amiben sok minden egymásra épül. A szakdolgozatomban a következő alprojekteken fogok változtatni: JSAN, és az erre épülő JSAN2Lim. Emellett még megtalálhatóak a következő alprojektek is: ESLintRunner, ESLint2Graph, LIM2Metrics, LIM2Patterns, DuplicatedCodeFinder és a ChangeTracker. Azért kell a JSAN2Limben is változtatni, mivel ha gyökerestül megváltoztatom a JSAN-t, akkor a JSAN2Lim rossz eredményeket fog visszaadni a typescript fileok elemzése közben.

A projekten amin fejlesztettem, azon többen is fejlesztettek egyszerre, ezáltal voltak átfedések, oszthatatlan részek. A projekt legtöbb része egymásra épül, ezért kiemelten fontos volt a csapatmunka egyes részeknél, hogy a projekt eredményesen és hatékonyan haladjon. Az eredményes csapatmunka magában foglalja az eredmények és az előrehaladás folyamatos kommunikálását, ami azt jelenti, hogy mindketten megértjük, hogy milyen célkitűzések vannak a projektben, és rendszeresen jelentést adunk egymásnak az elvégzett munkáról és az elért eredményekről.

Fejlesztés során úgynevezett JavaScriptSchema.vpp file szerkesztése is csoportos munka volt, hiszen minden erre épült, ez volt az alapja mindennek. Sok időt vett igénybe a szerkesztés. Nulláról kellett kellett újraírni az egész vpp filet. Sajnos dokumentáció nem készült az előző filehoz, ezért nekünk kellett kitalálni, hogy mi mit csinált, hiszen akik ezt

írták, ők már nem foglalkoztak ezzel. A szerkesztéshez szükséges volt a Visual Paradigm alkalmazást használni. Ezzel egyikőnk sem találkozott még, szóval először ezt kellett tanulmányozni, megérteni. Ezután értelmezni kellett a meglévő schemát, hiszen eddig az jól működött, csak nem lehetett könnyen bővíteni. Mérlegeltük a két opciót, ahol vagy megpróbáljuk bővíteni a jelenlegi schémát, vagy nulláról elkezdjük újraírni. Végül az újraírás mellett döntöttünk, hiszen ezt láttuk gyorsabb és könnyebb megoldásnak. Egy könnyen bővíthető, dokumentál schema volt az elképzelés. Ketten fejlesztettük le végül ezt a schemát.

JavaScriptSchema átírása

-mi ez -miért kellett átírni (JSAN megemlítése) -nehézségek, problémák (literal) schemában, kódkeresés, nem volt meg öröklődés, nulla dokumentáció -ábrák

1.1. JavaScriptSchemáról

1.2. Átírási okok

1.3. Nehézségek, problémák

JavaScriptAddon szerkesztése

-mit ad factory, mi ez a külön file -Miben változott

2.1. JavaScriptAddonról

2.2. Miben változott

JSAN2Lim átírása

-halstead, stb -miben kellett bővíteni, mi hiányzott -mit nem detektált

3.1. JSAN2Limről

3.2. Bővítések

3.3. Miket nem detektált

AST Binder Optimalizálás

-node addon mi -miért lassú -javítás, ha lehet, vagy ha nem, miért - nehézségek, eredmények

4.1. Binderről

4.2. Lassúság okai

4.3. Optimalizálás

4.4. Eredmény

Regteszt frissítés

jobb lett, működik hogy van tesztelve, output és performance vizsgálat TS tesztek hozzáadása / eredmények átnézése Specifikusabb tesztek hozzáadása (kapcsolókra tesztelünk -> JSAN / ESLintrunner 1 mappa / 1 kapcsoló) Python beli nehézségek tesztátírás során

5.1. Röviden a regtesztről

5.2. Tesztekről

5.3. Tesztek kibővítése

Összefoglaló

- javult a program - VPP up do datere hozása - JSAN további optimalizálása

6.1. Program javulása

6.2. Tervek

6.2.1. JavaScriptSchema naprakészre hozása

6.2.2. JSAN további optimalizálása

Nyilatkozat

Alulírott Pozsgai Alex programtervező informatikus BSc szakos hallgató, kijelentem, hogy a dolgozatomat a Szegedi Tudományegyetem, Informatikai Intézet Szoftverfejlesztés Tanszékén készítettem, programtervező informatikus BSc diploma megszerzése érdekében.

Kijelentem, hogy a dolgozatot más szakon korábban nem védtem meg, saját munkám eredménye, és csak a hivatkozott forrásokat (szakirodalom, eszközök, stb.) használtam fel.

Tudomásul veszem, hogy szakdolgozatomat a Szegedi Tudományegyetem Informatikai Intézet könyvtárában, a helyben olvasható könyvek között helyezik el.

Szeged, 2023. április 26.

.....
aláírás

Köszönetnyilvánítás