



IMPLEMENTATIELIJST_

Compiler APP

Dirk van Roosmalen

31 oktober 2024

INLEIDING

Dit document is bedoeld en wordt gebruikt als notitieblokje voor het benoemen van de features die geïmplementeerd zijn in mijn compiler. De aanwezigheid van een dergelijk bestand was vereist volgens de beschrijving van onderdeel 3. Inleveren in *ASSIGNMENT.md*.

1 EISEN WAAR HET PRODUCT AAN VOLDOET

1.1 Algemene eisen (4.1)

ID	Omschrijving	Prio	Punten	Voldoet
AL01	De code behoudt de packagestructuur van de aangeleverde startcode. Toegevoegde code bevindt zich in de relevante packages.	Must	0	Ja
AL02	Alle code compileert en is te bouwen met Maven 3.6 of hoger, onder OpenJDK 13. Tip: controleer dit door eerst mvn clean uit te voeren alvorens te compileren en in te leveren, hierop een onvoldoende halen is echt zonde. Gebruik van Oracle versies van Java is uitdrukkelijk niet toegestaan.	Must	0	Ja? JDK 17 is aanwezig in dit project.
AL03	De code is goed geformatteerd, zo nodig voorzien van commentaar, correcte variabelenamen gebruikt, bevat geen onnodig ingewikkelde constructies en is zo onderhoudbaar mogelijk opgesteld. (naar oordeel van docent)	Must	0	Ja
AL04	De docent heeft vastgesteld (tijdens les, assessment of op een andere manier) dat de compiler eigen werk is en dat je voldoet aan de beoordelingscriteria van APP-6, te weten: - Kent de standaardarchitectuur van compilers; - Kent de basisbegrippen over programmeertalen (zoals syntaxis, semantiek).	Must	0	Ja

1.2 Parseren (4.2)

ID	Omschrijving	Prio	Punten	Voldoet
PA00	De parser dient zinvol gebruik te maken van jouw eigen implementatie van een stack generiek voor ASTNode (VT: zie huiswerk IHANStack<ASTNode>)	Must	0	Ja
PA01	Implementeer een grammatica plus listener die AST's kan maken voor ICSS documenten die "eenvoudige opmaak" kan parseren, zoals beschreven in de taalbeschrijving. In level0.icss vind je een voorbeeld van ICSS code die je moet kunnen parseren. testParseLevel0() slaagt.	Must	10	Ja
PA02	Breid je grammatica en listener uit zodat nu ook assignments van variabelen en het gebruik ervan geparseerd kunnen	Must	10	Ja

	worden. In level1.icss vind je voorbeeldcode die je nu zou moeten kunnen parsen. testParseLevel1() slaagt.			
PA03	Breid je grammatica en listener uit zodat je nu ook optellen en aftrekken en vermenigvuldigen kunt parsen. In level2.icss vind je voorbeeld- code die je nu ook zou moeten kunnen parsen. Houd hierbij rekening met de rekenregels (vermenigvuldigen gaat voor optellen en aftrekken, optellen en aftrekken gaan van links naar rechts; zie ook deze site . testParseLevel2() slaagt.	Must	10	Ja
PA04	Breid je grammatica en listener uit zodat je if/else-statements aankunt. In level3.icss vind je voorbeeldcode die je nu ook zou moeten kunnen parsen. testParseLevel3() slaagt.	Must	10	Ja
PA05	PA01 t/m PA04 leveren minimaal 30 punten op	Must	0	Ja

1.3 Checken (4.3)

ID	Omschrijving	Prio	Punten	Voldoet
CH00	Minimaal vier van onderstaande checks moeten zijn geïmplementeerd	Must	0	Ja
CH01	Controleer of er geen variabelen worden gebruikt die niet gedefinieerd zijn.	Should	5	Ja
CH02	Controleer of de operanden van de operaties plus en min van gelijk type zijn. Je mag geen pixels bij percentages optellen bijvoorbeeld. Controleer dat bij vermenigvuldigen minimaal een operand een scalaire waarde is. Zo mag $20\% * 3$ en $4 * 5$ wel, maar mag $2px * 3px$ niet.	Should	5	Ja
CH03	Controleer of er geen kleuren worden gebruikt in operaties (plus, min en keer).	Should	5	Ja
CH04	Controleer of bij declaraties het type van de value klopt met de property. Declaraties zoals width: #ff0000 of color: 12px zijn natuurlijk onzin.	Should	5	Ja
CH05	Controleer of de conditie bij een if-statement van het type boolean is (zowel bij een variabele-referentie als een boolean literal)	Should	5	Ja
CH06	Controleer of variabelen enkel binnen hun scope gebruikt worden	Must	5	Ja

1.4 Transformeren (4.4)

ID	Omschrijving	Prio	Punten	Voldoet
TR01	Evalueer expressies. Schrijf een transformatie in Evaluator die alle Expression knopen in de AST door een Literal knoop met de berekende waarde vervangt.	Must	10	Ja

TR02	Evalueer if/else expressies. Schrijf een transformatie in Evaluator die alle IfClauses uit de AST verwijderd. Wanneer de conditie van de IfClause TRUE is wordt deze vervangen door de body van het if-statement. Als de conditie FALSE is dan vervang je de IfClause door de body van de ElseClause. Als er geen ElseClause is bij een negatieve conditie dan verwijder je de IfClause volledig uit de AST.	Must	10	Ja
------	--	------	----	----

1.5 Genereren (4.5)

ID	Omschrijving	Prio	Punten	Voldoet
GE01	Implementeer de generator in nl.han.ica.icss.generator.Generator die de AST naar een CSS2-compliant string omzet.	Must	5	Ja
GE01	Zorg dat de CSS met twee spaties inspringing per scopeniveau gegenereerd wordt.	Must	5	Ja

2 EIGEN UITBREIDINGEN (4.6)

2.1 Commentaar

De compiler is in staat om commentaar te verwerken als deze is aangegeven zoals bij CSS, door te beginnen met `(/*`) en te eindigen met `*/`). Dit commentaar wordt verwerkt en uiteindelijk bij de output weergegeven.

Commentaar kan zich voor, achter en tussen regels code bevinden bij variabele toewijzingen, stylerules en rule bodies. In onderstaand screenshot toon ik hoe dit praktisch gezien de gehele stylesheet omvangt. Ook wordt commentaar uit onware if-statements en niet-afgevuurde else statements verwijderd, zoals te zien bij *Comment8*.

ICSS Tool September 2020, version 1

File

Input (ICSS):

```

/* Comment1 */
AdjustColor := TRUE;
/* Comment2 */
p {
  /* Comment3 */
  background-color: #ffffff;
  if[AdjustColor] { /* Comment4 */
    color: #124532;
  }
  /* Comment5 */
}
/* Comment6 */
#menu {
  /* Comment7 */
  width: 20px;
  if[FALSE] {
    /* Comment8 */
    width: 10px;
  }
}

.menu {
  /* Comment9 */ color: #000000;
}

```

Internal (AST):

- ▼ Stylesheet
 - Comment
 - Comment
 - ▼ Stylerule
 - TagSelector p
 - Comment
 - ▼ Declaration
 - Property: (background-color)
 - Color literal (#ffffff)
 - Comment
 - ▼ Declaration
 - Property: (color)
 - Color literal (#124532)
 - Comment
 - Comment
 - ▼ Stylerule
 - IdSelector #menu
 - Comment
 - ▼ Declaration
 - Property: (color)
 - Color literal (#000000)

Output (CSS):

```

/* Comment1 */
/* Comment2 */
p {
  /* Comment3 */
  background-color: #ffffff;
  /* Comment4 */
  color: #124532;
  /* Comment5 */
}
/* Comment6 */
#menu {
  /* Comment7 */
  width: 20px;
}

.menu {
  /* Comment9 */
  color: #000000;
}

```

Pipeline:

Parse

Check

Transform

Generate

Generating output...

Generating succeeded

OPEN UP
NEW HAN_ UNIVERSITY
HORIZONS. OF APPLIED SCIENCES