

Bijlage Compiler eisen

Tessa van den Heuvel

2103174

ITN-CNI-B

Michel Koolwaaij

Versie 1

1. Algemeen

De aangeleverde code is gebaseerd op de startcode van ICSSTool en voldoet aan alle algemene eisen:

ID	Omschrijving	Prio	Punten	C V
AL01	De code behoudt de packagestructuur van de aangeleverde startcode. Toegevoegde code bevindt zich in de relevante packages.	Must	0	A
AL02	Alle code compileert en is te bouwen met	Must	0	n

ID	Omschrijving	Prio	Punten	C V
	<p>Maven 3.6 of hoger, onder OpenJDK 13.</p> <p>Tip: controleer dit door eerst <code>mvn clean</code> 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.</p>			
AL03	<p>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)</p>	Must	0	n

ID	Omschrijving	Prio	Punten	C V
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	A

2. Parser (PA)

De parser is geïmplementeerd conform de ICSS-20-SEP specificatie.

ID	Omschrijving	Prio	Punten
PA00	De parser dient zinvol gebruik te maken van jouw eigen implementatie van een stack generic voor ASTNode (VT: zie huiswerk IHANStack<ASTNode>)	Must	0
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

ID	Omschrijving	Prio	Punten
PA02	<p>Breid je grammatica en listener uit zodat nu ook assignments van variabelen en het gebruik ervan geparseerd kunnen worden. In <code>level1.icss</code> vind je voorbeeldcode die je nu zou moeten kunnen parseren.</p> <p><code>testParseLevel1()</code> slaagt.</p>	Must	10
PA03	<p>Breid je grammatica en listener uit zodat je nu ook optellen en aftrekken en vermenigvuldigen kunt parseren. In <code>level2.icss</code> vind je voorbeeld- code die je nu ook zou moeten kunnen parseren. Houd hierbij rekening met de rekenregels</p>	Must	10

ID	Omschrijving	Prio	Punten
	(vermenigvuldigen gaat voor optellen en aftrekken, optellen en aftrekken gaan van links naar rechts; zie ook deze site .” <code>testParseLevel2()</code> slaagt.		
PA04	Breid je grammatica en listener uit zodat je if/else-statements aankunt. In <code>level3.icss</code> vind je voorbeeldcode die je nu ook zou moeten kunnen parseren. <code>testParseLevel3()</code> slaagt.	Must	10
PA05	PA01 t/m PA04 leveren minimaal 30 punten op	Must	0

Opmerking: De parser is volledig operationeel volgens de tests `testParseLevel0()` t/m

testParseLevel3() .

3. Checker (CH)

De semantische checks zijn allemaal geïmplementeerd:

ID	Omschrijving	Prio	Punten	Comp VT
CH00	Minimaal vier van onderstaande checks moeten zijn geïmplementeerd	Must	0	APP-2 6, APP
CH01	Controleer of er geen variabelen worden gebruikt die niet gedefinieerd zijn.	Should	5	
CH02	Controleer of de operanden van de operaties plus en min van gelijk type zijn. Je mag geen pixels bij percentages	Should	5	

ID	Omschrijving	Prio	Punten	Comp VT
	optellen bijvoorbeeld. Controleer dat bij vermenigvuldigen minimaal een operand een scalaire waarde is. Zo mag <code>20% * 3</code> en <code>4 * 5</code> wel, maar mag <code>2px * 3px</code> niet.			
CH03	Controleer of er geen kleuren worden gebruikt in operaties (plus, min en keer).	Should	5	
CH04	Controleer of bij declaraties het type van de value klopt met de property. Declaraties zoals <code>width: #ff0000</code>	Should	5	

ID	Omschrijving	Prio	Punten	Comp VT
	of <code>color: 12px</code> zijn natuurlijk onzin.			
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	
CH06	Controleer of variabelen enkel binnen hun scope gebruikt worden	Must	5	

De tests uit `CheckerTest.java` bevestigen dat de checks correct werken.

4. Evaluator / Transformatie (TR)

De evaluator voldoet aan alle eisen:

ID	Omschrijving	Prio	Punten	Competen VT
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	APP-2, AP 6, APP-7
TR02	Evalueer if/else expressies. Schrijf een transformatie in Evaluator die alle IfClause s	Must	10	APP-2, AP 6, APP-7

ID	Omschrijving	Prio	Punten	Competen VT
	<p>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</p>			

ID	Omschrijving	Prio	Punten	Competen VT
	verwijder je de IfClause volledig uit de AST.			

Hierdoor wordt de AST volledig getransformeerd en klaar voor generatie.

5. Generator (GE)

De generator voldoet aan alle eisen:

ID	Omschrijving	P
GE01	Implementeer de generator in nl.han.ica.icss.generator.Generator die de AST naar een CSS2-compliant string omzet.	M
GE02	Zorg dat de CSS met twee spaties inspringing per scopeniveau gegenereerd wordt.	M

6. Eigen uitbreidingen

- Geen eigen taaluitbreiding geïmplementeerd in deze versie.
- Alle punten uit de verplichte opdrachten zijn behaald, echter extra functionaliteit zoals boolean expressies of optimalisaties is niet toegevoegd.

7. Conclusie

De ingeleverde code voldoet aan alle Must- en should-eisen van de opdracht, met uitzondering van de eigen uitbreidingen. Alle tests voor parser, checker, evaluator en generator slagen correct. De code is onderhoudbaar en volgt de door de docent vastgestelde structuur en richtlijnen.