

## Algemene eisen

ID	Omschrijving	Prio	Punten
AL01	De code behoudt de packagestructuur van de aangeleverde startcode. Toegevoegde code bevindt zich in de relevante packages.	Must	0
AL02	Alle code compileert en is te bouwen met Maven 3.6 of hoger, onder OpenJDK 13. Tip: controleer dit door <b>eerst</b> <code>mvn clean</code> uit te voeren alvorens te compileren en in te leveren. <b>Gebruik van Oracle versies van Java is uitdrukkelijk niet toegestaan.</b>	Must	0
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

Er wordt gebruik gemaakt van OpenJDK13 van <https://openjdk.java.net/>.

In figuur 1 is te zien een succesvol uitslag van de `mvn clean` command.

```
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ icsstool ---
[INFO] Deleting C:\Users\IMila\OneDrive\Bureaublad\DreamBot\icss2020\startcode\target
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.511 s
[INFO] Finished at: 2020-03-21T23:00:47+01:00
```

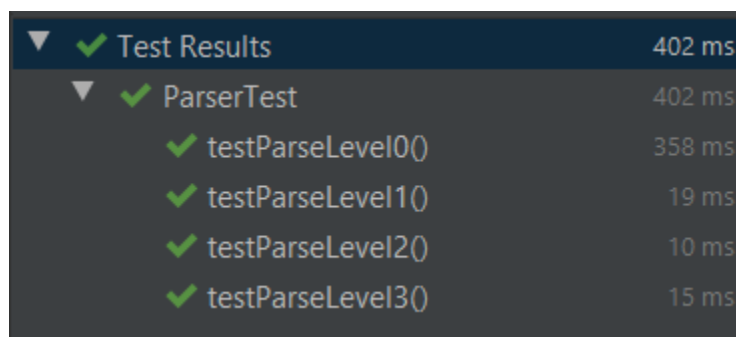
Figuur 1

De code is volgens mijn eigen oordeel goed geformatteerd, is voorzien van correcte variabelenamen en bevat geen onnodige ingewikkelde constructies.

## Parseren (40 punten)

ID	Omschrijving	Prio	Punten
PA01	Implementeer een parser plus listener die AST's kan maken voor ICSS documenten die "eenvoudige opmaak" kan parseren, zoals beschreven in de taalbeschrijving. In <code>level0.icss</code> vind je een voorbeeld van ICSS code die je moet kunnen parseren. <code>testParseLevel0()</code> slaagt.	Must	10
PA02	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. <code>testParseLevel1()</code> slaagt.	Must	10
PA03	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. <code>testParseLevel2()</code> slaagt.	Should	10
PA04	Breid je grammatica en listener uit zodat je if-statements aankunt. In <code>level3.icss</code> vind je voorbeeldcode die je nu ook zou moeten kunnen parseren. <code>testParseLevel3()</code> slaagt.	Should	10

Hieronder bevindt zich een figuur waar te zien is dat alle unit tests geslaagd zijn. Hiermee kan men een aanname maken dat alle eisen zijn behaald van het gedeelte van parseren.



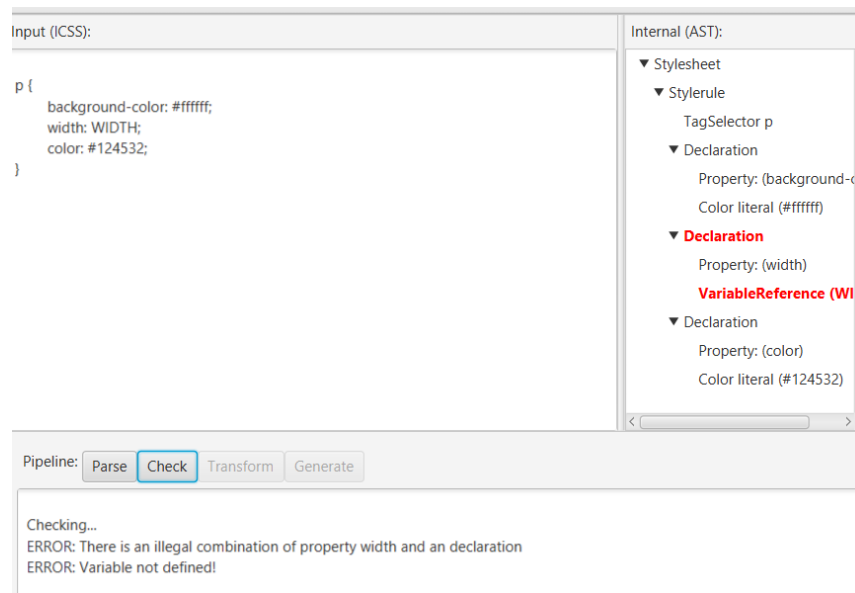
▼ ✓ Test Results	402 ms
▼ ✓ ParserTest	402 ms
✓ testParseLevel0()	358 ms
✓ testParseLevel1()	19 ms
✓ testParseLevel2()	10 ms
✓ testParseLevel3()	15 ms

Figuur 2

## Checken (20 punten)

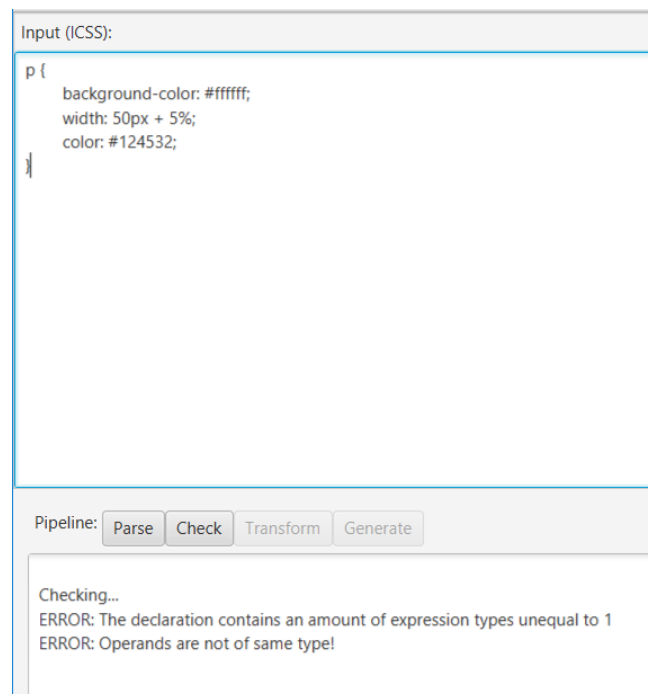
ID	Omschrijving	Prio	Punten
CH00	Minimaal drie van onderstaande checks <b>moeten</b> zijn geïmplementeerd	Must	0
CH01	Controleer of er geen variabelen worden gebruikt die niet gedefinieerd zijn.	Should	4
CH02	Controleer of de operanden van de operaties plus en min van gelijk type zijn en dat vermenigvuldigen enkel met scalaire waarden gebeurt. Je mag geen pixels bij percentages optellen bijvoorbeeld.	Should	4
CH03	Controleer of er geen kleuren worden gebruikt in operaties (plus, min en keer).	Should	2
CH04	Controleer of bij declaraties het type van de value klopt met de property. Declaraties zoals <code>width: #ff0000</code> of <code>color: 12px</code> zijn natuurlijk onzin.	Should	2
CH05	Controleer of de conditie bij een if-statement van het type boolean is (zowel bij een variabele-referentie als een boolean literal)	Should	4
CH06	Controleer of variabelen enkel binnen hun scope gebruikt worden	Should	4

CH01 controleert of een variabele wordt gebruikt die niet gedefinieerd is. In het figuur hieronder bevindt zich een stuk CSS code waarin een variabele niet is gedefinieerd. Hierdoor krijg je een foutmelding met daarin “Variabel not defined”. Hierdoor is CH01 geslaagd.



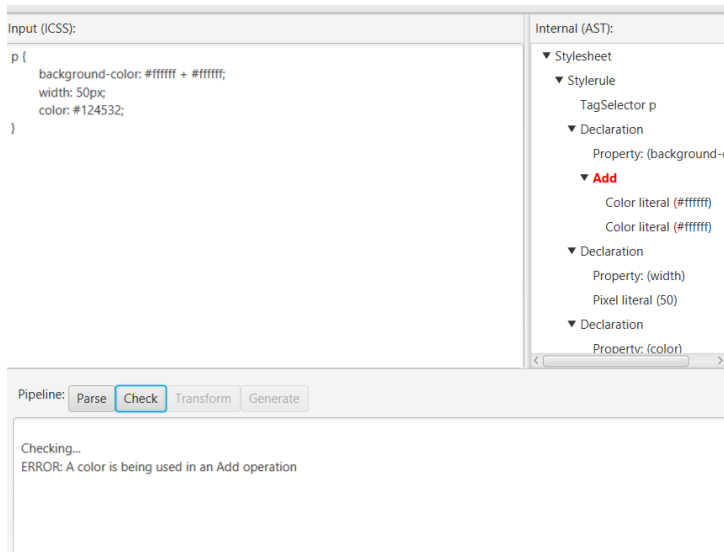
Figuur 3

Bij CH02 worden de operanden gecontroleerd of ze van hetzelfde type zijn. In het figuur hieronder bevindt zich een stuk code dat twee verschillende typen operanden gebruikt waardoor check een foutmelding geeft. Hierdoor is CH02 behaald.



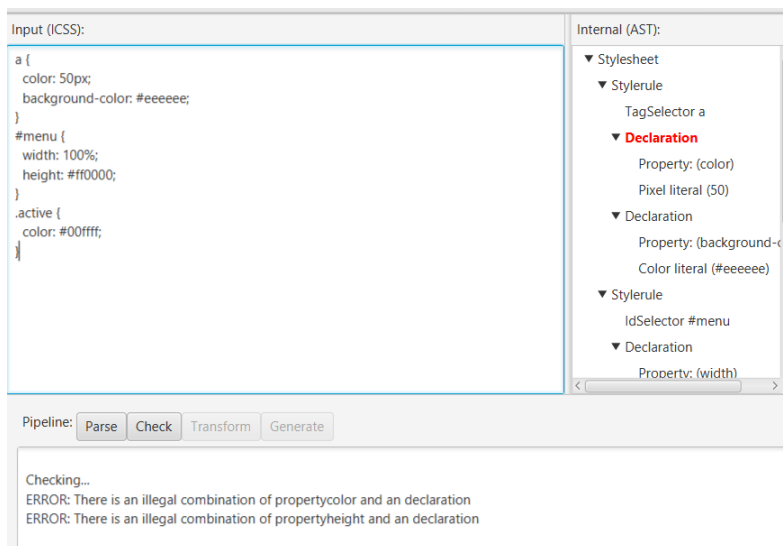
Figuur 4

CH03 houdt in dat hex waarden niet gebruikt kan worden in operaties. In het onderstaande figuur ziet u een test waarin hex waarden worden opgeteld. De code is getest met alle operaties. CH03 is hierdoor behaald



Figuur 5

Om CH04 te testen, is er CSS code gebruikt waarvan de waarden van property color en width zijn omgewisseld. Zoals te lezen is in het figuur, krijg je bij het checken voor elke verkeerde type waarde een foutmelding. Hierdoor is CH04 behaald.



Figuur 6

CH05 houdt in dat een conditie in een if statement van een boolean type is. Om dit te testen wordt ParWidth als een boolean gebruikt bij een if expressie. In het figuur hieronder is te lezen dat hierdoor een foutmelding ontstaat namelijk: "A non boolean variable is being used as boolean in if statement: PIXEL". Hierdoor is CH05 behaald.

**Input (ICSS):**

```
ParWidth := 50px;
AdjustColor := TRUE;
LinkColor := #000000;

p {
  background-color: #ffffff;
  width: ParWidth;
  if [ParWidth] {
    color: #124532;
    if [FALSE]{
      background-color: LinkColor;
    }
  }
}
```

**Internal (AST):**

- ▼ Declaration
  - Property: (width)
  - VariableReference (ParV
- ▼ **If\_Clause**
  - VariableReference (ParV
- ▼ Declaration
  - Property: (color)
  - Color literal (#124532
- ▼ If\_Clause
  - Bool Literal (FALSE)
- ▼ Declaration
  - Property: (backgrou
  - VariableReference (

**Pipeline:** Parse Check Transform Generate

Checking...  
ERROR: A non boolean variable is being used as boolean in if statement: PIXEL

Figuur 7

## Transformeren (20 punten)

ID	Omschrijving	Prio	Punten
TR01	Implementeer de <code>EvalExpressions</code> transformatie. Deze transformatie vervangt alle <code>Expression</code> knopen in de AST door een <code>Literal</code> knoop met de berekende waarde.	Should	10
TR02	Implementeer de <code>RemoveIf</code> transformatie. Deze transformatie verwijdert alle <code>IfClauses</code> uit de AST. Wanneer de conditie van de <code>IfClause</code> <code>TRUE</code> is wordt deze vervangen door de body van het if-statement. Als de conditie <code>FALSE</code> is dan verwijder je de <code>IfClause</code> volledig uit de AST.	Should	10

TR01 en TR02 zijn beide geïmplementeerd. Zie code.

## Genereren (10 punten)

ID	Omschrijving	Prio	Punten
GE01	Implementeer de generator in nl.han.ica.icss.generator.Generator die de AST naar een CSS2-compliant string omzet.	Must	5
GE02	Zorg dat de CSS met twee spaties inspringing per scopeniveau gegenereerd wordt.	Must	5

De ASTNode is aangepast dat hij een `getValue()` functie. Deze wordt geërfd de subklassen van `Literal`. Deze hebben een methode met override om de waarde van de literal op te halen. Dit wordt gebruikt bij het genereren van de CSS code.

Het figuur hieronder bevat een stuk ICSS code die geconverteerd wordt naar CSS2 code. Hieraan zie je dat het voldoet aan GE01 en GE02.

Input (ICSS):	Internal (AST):	Output (CSS):
<pre>ParWidth := 50px; AdjustColor := TRUE; LinkColor := #000000;  p {   background-color: #ffffff;   width: ParWidth;   if [TRUE] {     color: #124532;     if [FALSE]{       background-color: LinkColor;     }   } }</pre>	<pre>▼ Stylesheet   ▼ VariableAssignment (ParWid     VariableReference (ParWid       Pixel literal (50)   ▼ VariableAssignment (AdjustC     VariableReference (AdjustC       Bool Literal (TRUE)   ▼ VariableAssignment (LinkCol     VariableReference (LinkCo       Color literal (#000000)   ▼ Stylerule     TagSelector p   ▼ Declaration</pre>	<pre>p {   background-color: #ffffff;   width: 50px;   color: #124532; }</pre>

Figuur 8

## Eigen uitbreidingen (10 punten)