



Compiler and OS :

—The real deal compiler

Nathan Ganesh - 489290

Saxion deventer

DHI2S0

Index page

Index page	2
Introduction	2
Requirements	3
Questions to be answered.	3
TheRealDeal language	3
Data types	3
Error types	4
Declaration of a variable and scopes	4
Errors with variables(bad weather)	5
Functions	5
Errors with functions	6
Ask for user input	7
Expressions	7
Functions with expressions	9
if/else & while statement	9
Design choices	10
Testing	10
Dev issues	10

Introduction

Throughout the year a lot of different programming languages have been created. For the coming weeks you will build your own compiler for a programming language you devised yourself. The compiler should be able to parse programs in your language, check if they are correct and subsequently transform them into Java Bytecode. This way, programs in your own language can be run on any system with the Java Virtual Machine installed.

This document will be describing the self made programming language called therealdeal programming language.

Requirements

The following requirements had to be minimal be met:

1. At least 4 data types.
2. Support for variables.
3. Arithmetic Expressions.
4. Logical expressions.
5. Conditional branching (eg: if statements).
6. Looping statements (while / for).
7. Function with parameters and return type.
8. Support for global variables.
9. Support for printing expressions and variables to consoles.
10. Support for reading user input.

Questions to be answered.

Is your language an esoteric (joke) language or is a well-useable language?

What would branching look like?

Are you supporting loops?

What do functions look like?

Is a programmer able to use comments and what is their syntax? What extra feature will you add?

TheRealDeal language

TheRealDeal language is made with simplicity in mind. These days people overcomplicate things they are trying to save the world from itself. The main philosophy about this language is that it will have an emphasis on the importance of non-action/non-resistance, "going with the

flow” to write an elevated and transformative code. This will be a mix of javascript non static typed language with java typing. Something like javascript + typescript combination.

Note that some parts are capitalized because of word autoformatting.

Data types

The 4 chosen data types are int, double string and boolean.

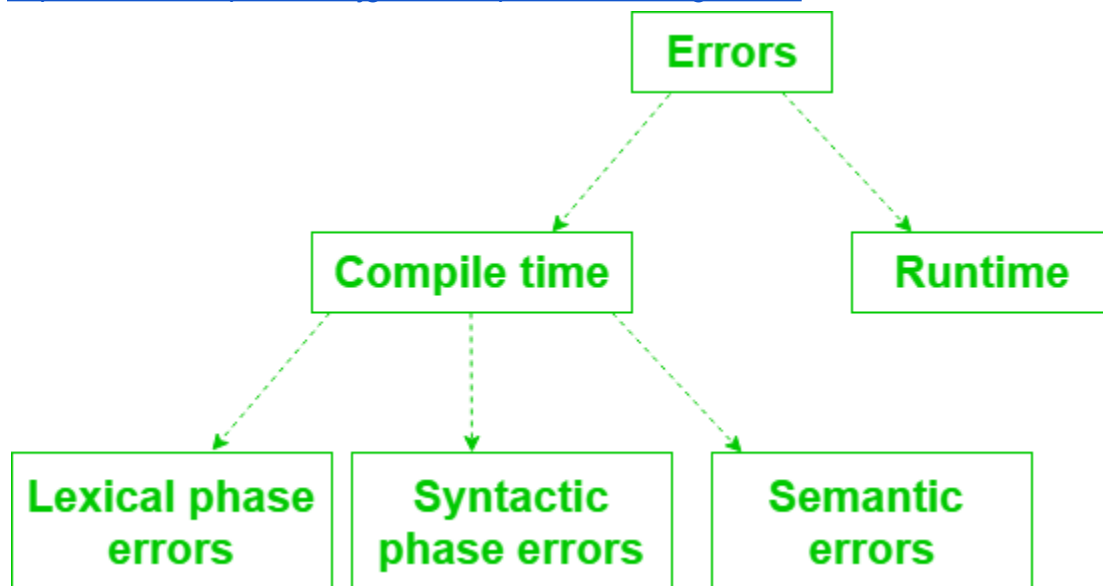
Data type	Name in code	Example
int	int	int basicInt = 5;
double	double	double basicDouble=5.0;
string	Text	Text basicText = “oek”
boolean	boolean	boolean basicBoolean = true;

Error types

There are different types of error during the writing of the code.

This source is used for finding different types of errors.

<https://www.mindprod.com/jgloss/compileerrormessages.html>



Declaration of a variable and scopes

The variables declared outside a function are static and can be called in every scope. In scope variables in the function have the highest priority.

Good weather cases

Explanation	Code
Init global scope variable	<code>int oek = 10;</code>
Init second variable. Defaults to 0 also global scope	<code>Int otherOek;</code>
Create oekOek()	<code>Void oekOek(){ Int oek = 5; print(oek) //print 5 print(otherOek) //print 0; }</code>

Errors with variables(bad weather)

List of possible errors with variables.

Error	Example
init text variable with OEK	<code>Text oek = "OEK";</code>
Missing variable initializers	<code>Snoek = "oh"</code>
Undefined variable(non type)	<code>Oek oek2 = 5.0;</code>
Duplicate variable	<code>Int oek = 5;</code>
Variable not found	<code>Double koek = aap+oek;</code>
Type doesn't match	<code>Text bigText = true;</code>

Functions

There are 4 different functions, int, double,boolean,void and text. In this part different examples will be created of functions.

Good weather

explanation	Code
Create a function that adds 2 values together.	<code>int oekPlusOek(int oek1, int oek2){</code>

	<pre> return oek1+oek2; } </pre>
Boolean function with function parameters.	<pre> Boolean oekBoolean(int age){ if(age<18){ print("Oek oek"); return true; }else{ print("No Oek oek for you"); return false; } } </pre>
Int function	<pre> int oekMoney(){ return 999;} </pre>
Void no return type.	<pre> void greet(){ print("welcome to oekscape"); } </pre>
Double function	<pre> double doubleOek(){ return 9.9; } </pre>
Functional programming	<pre> Void functionalOekEqual(){ if(5 == 5){ print(5); return; } print(5050); } </pre>
Text function	<pre> Test oekText(int age){ if(age<18){ return "Oek oek"; }else{ return "No Oek oek for you"; } } </pre>

Errors with functions

Bad weather cases.

Error	Example
Function not found.	<pre>Void oekToSaurus(){ print("oek+dinosaurus = oekToSaurus"); } oek();</pre>
Dogshit return type.	<pre>Void oek(){ Return "oek*oek = 2 Oek" }</pre>
Duplicate parameter	<pre>Text oek(Text oek, Text oek){ Return oek+oek; }</pre>
Unreachable code	<pre>Void oekOek(){ Return; print("oekOek"); }</pre>

Ask for user input

Good weather cases of core functionality.

- Requirement: There should be a possibility to ask a user for a value, at least for one single data type.

explanation	code
Ask int.	<code>Int oek = askInt();</code>
Ask double	<code>Double oek = askDouble();</code>
Ask boolean	<code>Boolean oek = askBoolean();</code>
Ask text	<code>Text oek = askText();</code>

It will obviously throw an error if you put a text value into an int.

Expressions

A Java expression consists of variables, operators, literals, and method calls. To know more about method calls, visit Java methods. For example, ... Here, `number1 == number2` is an expression that returns a boolean value. There are a couple of requirements to meet.

http://www.cs.bilkent.edu.tr/~guvenir/courses/CS101/op_precedence.html

Java Operator Precedence Table

Precedence	Operator	Type	Associativity
15	() [] .	Parentheses Array subscript Member selection	Left to Right
14	++ --	Unary post-increment Unary post-decrement	Right to left
13	++ -- + - ! ~ (type)	Unary pre-increment Unary pre-decrement Unary plus Unary minus Unary logical negation Unary bitwise complement Unary type cast	Right to left
12	* / %	Multiplication Division Modulus	Left to right
11	+ -	Addition Subtraction	Left to right
10	<< >> >>>	Bitwise left shift Bitwise right shift with sign extension Bitwise right shift with zero extension	Left to right
9	< <= > >= instanceof	Relational less than Relational less than or equal Relational greater than Relational greater than or equal Type comparison (objects only)	Left to right
8	== !=	Relational is equal to Relational is not equal to	Left to right
7	&	Bitwise AND	Left to right
6	^	Bitwise exclusive OR	Left to right
5		Bitwise inclusive OR	Left to right
4	&&	Logical AND	Left to right
3		Logical OR	Left to right
2	? :	Ternary conditional	Right to left
1	= += -= *= /= % =	Assignment Addition assignment Subtraction assignment Multiplication assignment Division assignment Modulus assignment	Right to left

Larger number means higher precedence.

explanation	code
Simple boolean + assignment	Boolean oek = true;
Simple +	Int oek = 3+3;
Simple -	Int oek = 19-20;

Simple *	Int oek = 3*3;
Simple /	Int oek =3/3;
() → precedence rules	Int oek = 1*(1+1)+1
Using variables	Int k = 3; Int x = k * 3;
Comparison boolean assignment	boolean oek = 20 == 20;

Functions with expressions

An expression should always return something if nothing gets returned it should throw an error.

Error	Example
Create a test function(global scope)	int oekPlusOek(int oekInt){ Return oekInt + 5; }
Adding value together from other function	int oekP(){ Return oekPlusOek(123) + 5; }
Double function	double oekDouble(){ Return 1.0*5.0; }
Boolean expression return	boolean oek(boolean oekBool){ Return oekBool; }
Complex boolean expression	Boolean y(int oek1, int oek2){ Return oek(oek1==oek2); }

if/else & while statement

The while statement is used for looping.

The if statement is used to check for branching.

explanation	code
-------------	------

Infinite loop anything that can result in a true statement in a while loop can make it execute.	<pre>while(1==1){ print("oek oek!!") }</pre>
If statements work the same as a while loop. If you can make it result in a true it will execute the code inside the block.	<pre>if(5!=999){ print("well OEKOEK");}</pre>
If else block	<pre>if(5==292){ }else{ print("WELL no oekoek"); }</pre>

Design choices

Testing

http://www.cs.bilkent.edu.tr/~guvenir/courses/CS101/op_precedence.html

Dev issues

Write a number of example programs in your language that demonstrate the features. You can use these example programs as tests for your compiler later. Additionally, create a document that describes how to use all the features in your programming language and how a programmer should use them. You can create a Word/PDF-document for this or use Markdown and add it to your compiler repository

Deliverables

- A description of your language
- Example programs:
 - At least 5 working examples
 - At least 5 programs demonstrating compiler errors
- Unit tests, for both correct and incorrect code
- A compiler, supporting at least:
 - Variables and different data types
 - Calculating expressions: +, -, *, /
 - Logic expressions: ==, !=, and, or
 - Some kind of loop
 - Some kind of branching mechanism
 - Printing, user input
 - Functions or subroutines

See the
assignment

The assignment asks you to make a number of choices. Is your language an esoteric (joke) language or is a well-useable language? What would branching look like? Are you supporting loops? What do functions look like? Is a programmer able to use comments and what is their syntax? What extra feature will you add?

