

KIDEB
Onze eigen programmeertaal

H.M. Bastiaan
s1204254
Doctor van Damstraat xx

V.J. Smit
s1206257
Kremersmaten 168

3 juli 2014

Inhoudsopgave

1	Inleiding	2
2	Taalbeschrijving	3
3	Problemen & oplossingen	4
4	Syntax, context & semantiek	5
4.1	Syntax	5
4.1.1	Terminale symbolen	5
4.1.2	Non-terminale symbolen	5
4.1.3	Productieregels	6
4.2	Context	6
4.3	Semantiek	6
5	Vertaalregels	7
6	Java programmatuur	8
7	Testplan & resultaten	9
7.1	Testen	9
7.2	Resultaten	9
8	Conclusies	10
9	Appendix	11
9.1	Lexer specificatie	11
9.2	Parser specificatie	15
9.3	Parser specificatie	15
9.4	In- en uitvoer testprogramma	16

Hoofdstuk 1

Inleiding

Hoofdstuk 2

Taalbeschrijving

KIDEB is een kleine, imperatieve programmeertaal met beperkte mogelijkheden, ontwikkeld als project voor het vak vertalerbouw als onderdeel van de Technische Informatica bachelor van de Universiteit Twente. Ondanks de beperkte mogelijkheden bevat de taal toch enkele leuke onderdelen.

Uiteraard is de basis van een programmeertaal ook aanwezig in KIDEB. Variabelen kunnen worden gedeclareerd met primitieve typen: integer, boolean en character. Door toegevoegde type inferentie hoeft het primitieve type zelfs niet expliciet te worden vermeld.

Op deze variabelen zijn een aantal bewerkingen mogelijk. Zo zijn de standaard boolean operaties beschikbaar, om verschil of gelijkheid te bepalen. Daarnaast bevat de taal de basis rekenkundige operaties en zelfs bestaat de mogelijkheid tot machtsverheffen.

Belangrijke statements zijn ook geïmplementeerd. If-else is onderdeel van de taal, evenals een while-statement. Voor het doorlopen van een complete array is het for-statement bijgevoegd.

De eerste belangrijke uitbreiding is de mogelijkheid gebruik te maken van subroutines. Naast het hoofdprogramma zijn namelijk ook functies te definiëren en uit te voeren. Functies hebben de mogelijkheid tot het opleveren van een waarde, maar dat is niet verplicht.

De tweede belangrijke uitbreiding was al even genoemd. Dit is namelijk het gebruik van arrays. Door het gebruik van arrays wordt de taal een stuk complexer, maar ook een stuk sterker.

De volledige grammatica van de taal is te vinden in het hoofdstuk 4.

Hoofdstuk 3

Problemen & oplossingen

Hoofdstuk 4

Syntax, context & semantiek

4.1 Syntax

Deze sectie beschrijft de symbolen en productieregels van KIDEB. Samen vormen deze de totale grammatica van de taal.

4.1.1 Terminale symbolen

De terminale symbolen:

-

4.1.2 Non-terminale symbolen

De non-terminale symbolen:

program (startsymbool)

command

declaration

- var_declaration
- scope_declaration
- func_declaration

assignment

- var_assignment

argument

- arguments

statement

- while_statement
- if_statement
- if_part

- else_part
- for_statement
- return_statement
- assign_statement

expression

- expressionAO
- expressionLO
- expressionPM
- expressionMD
- expressionPW
- expression_list
- call_expression operand
- array_literal
- array_value_list

type

- primitive_type
- compositie_type

identifier

number

4.1.3 Productieregels

4.2 Context

4.3 Semantiek

Hoofdstuk 5

Vertaalregels

Hoofdstuk 6

Java programmatuur

Hoofdstuk 7

Testplan & resultaten

7.1 Testen

7.2 Resultaten

Hoofdstuk 8

Conclusies

Hoofdstuk 9

Appendix

9.1 Lexer specificatie

Voor de lexer zijn de verschillende tokens van belang. Deze tokens staan hieronder allen gedefinieerd.

Tekens

Token	teken
COLON	:
SEMICOLON	;
LPAREN	(
RPAREN)
LBLOCK	[
RBLOCK]
LCURLY	{
RCURLY	}
COMMA	,
DOUBLE_QUOTE	”
SINGLE_QUOTE	\
BODY	body

Operators

Token	teken
PLUS	+
MINUS	-
DIVIDES	/
MULTIPL	*
POWER	^
LT	<
GT	>
GTE	>=
LTE	<=
EQ	=
NEQ	!
ASSIGN	==
OR	
AND	&&

Keywords van KIDEB

Token	keyword
PROGRAM	program
SWAP	swap
IF	if
THEN	then
ELSE	else
DO	do
WHILE	while
FROM	from
IMPORT	import
BREAK	break
CONTINUE	continue
RETURN	return
FOR	for
IN	in
RETURNS	returns
FUNC	func
ARRAY	array
ARGS	args
VAR	var
OF	of

Standaard types

Token	keyword
INTEGER	int
CHARACTER	char
BOOLEAN	bool
CALL	call

9.2 Parser specificatie

9.3 Parser specificatie

9.4 In- en uitvoer testprogramma