TDP019 -

Jonas Malmlöf (jonma993)
Denis Ivan Blazevic (denbl369)

BNF Tree

```
<Program>            ::= 'create program' <statements> 'end program'
<statements>         ::= <statement> <statements> | <empty>
<statement>          ::= <create>
                       | <modify>             ## ? add, multiply, subtract, divide
                       | <call>
                       | <set>
                       | <conditional>
                       | <input>
                       | <output>
<create>             ::= 'create' <integervar>
                       | 'create' <stringvar>
                       | 'create' <boolvar>
                       | 'create' <doublevar>
                       | 'create function' <funct_name>
<integervar>         ::= 'variable/int'                       ## ?
<stringvar>          ::= 'variable/string'                    ## ?
<boolvar>            ::= 'variable/bool'                      ## ?
<doublevar>          ::= 'variable/double'                    ## ?
<funct_name>         ::= <variable> [a-zA-Z]                  ## ?
<call>               ::= 'call' <func_name>
<set>                ::= 'set' <variable> 'to' <variable>     ## ?
<modify>             ::= 'add' <digit> 'to' <variable>
                       | 'subtract' <digit> 'from' <variable>
                       | 'multiply' <variable> 'with' <digit>
                       | 'divide' <variable> 'by' <digit>
<conditional>        ::=  <if_statement>
                     | <or_statement>
                     | <else_statement>
                     | <loop_statement>
<if_statement>       ::= 'if' <greater_condition> 'then'
                       | 'if' <greater_condition> 'then' <statements> 'else'
<loop_statement>     ::= 'loop until' <greater_condition> <statements> 'end loop'
<greater_condition>  ::= <variable> <condition> <variable>
<condition>          ::= 'is greater than'
                       | 'is smaller than'
                       | 'is equal to'
                       | 'is not equal to'             ## ?
<input>              ::= 'input to' <variable>                ## ?
<output>             ::= 'print' <variable>
                       | 'print' <digit>
<variable> ::=                                                ## ?
```

```
<digit>  ::= [0-9+]
<string> ::= [a-z A-Z+]

<expression> ::= ## ?
<equation>   ::= ## ? its own built-in function.
<set> = <assign> ??
```