**Tiny Language Regular Expressions**

# Basic Standards

Letter:=[\_|a-z|A-Z] Digit:=[0-9]

Boolean\_Operator:=(&&|\|\|)

Stats\_set:=(Assignment\_Stat|Write\_Stat|Read\_Stat|Return\_Stat|Declaration\_Stat)

# Tiny RegEx:

Number := (Digit)+(\.(Digit+))?

String := “(Letter|digit|A.operator)\*”

Res\_Keywords := (int|float|string|read|write|repeat|until|if|elseif|else|return|endl)

Comment := /\\*String\\*/

Identifier := Letter(Letter|Digit)\*

Function\_Call := Identifier\(Identifier(,Identifier\*)?\);

Term := Number|Identifier|Function\_Call

Arethmetic\_Operator := \+|-|\\*|/

Equation := ((\(Term Arethmetic\_Operator Term\))+ | (Term Arethmetic\_Operator Term)+)+

Expression := (String | Term | Equation)

Assignment\_Stat := Identifier \:= Expression;

Datatypes := (int|float|string)

Declaration\_Stat := Datatype (Identifier |Assignment\_stat) (\, Identifier|Assignment\_stat)\*

Write\_Stat := write (Expression|endl);

Read\_State := read Identifier;

Return\_Stat := return Expression;

Condition\_Operator := (<|>|=|<>)

Condition := Identifier Condition\_Operator Term

Condition\_Stat := Condition ((Boolean\_Operator Condition)+)?

If\_Stat := if Condition\_Stat then Stats\_set\* (ElseIF\_Stat|Else\_Stat)\* end

ElseIf\_Stat := elseif Condition Stat then Stats\_set\* (ElseIF\_Stat|Else\_Stat)\* end

Else\_Stat := else Stats\_set\* end

Repeat\_Stat := repeat Stats\_set until Condition\_Stat

FunctionName := Identifier

Parameter := Datatype Identifier

Function\_Declaration := Dataype FunctionName \(Parameter (,Parameter)\*\)

Function\_Body := \{ Stats\_set\* Return\_Stat\}

Function\_Stat := Function\_Declaration Function\_Body

Main\_Function := Datatype main \(\) Function\_Body

Program := Function\_Stat\* Main\_Function