



# **Design of Compilers**

## **Parser User Guide**

By:

**Dalia Ayman Ahmed**

CESS pre-junior

Presented to:

**Dr. Sahar Haggag**

**Eng. Amr Saad**

The Parser receives the tokens from the scanner using the function *getToken()*; reading from the file “test.txt”.

According to the grammatical rules in EBNF:

```
program -> stmt-sequence
stmt-sequence -> stmt {; stmt}
stmt -> if-stmt | repeat-stmt | assign-stmt | read-stmt | write-stmt
if-stmt -> if exp then stmt-sequence [ else stmt-sequence] end
repeat-stmt -> repeat stmt-sequence until exp
assign-stmt -> identifier := exp
read-stmt -> read identifier
write-stmt -> write exp
exp -> simple-exp [comparison-op simple-exp]
comparison-op -> < | =
simple-exp -> term {addop term}
addop -> + | -
term -> factor {mulop factor}
mulop -> * | /
factor -> (exp) | number | identifier
```

The Parser checks whether the input text matches the grammatical rules or not.

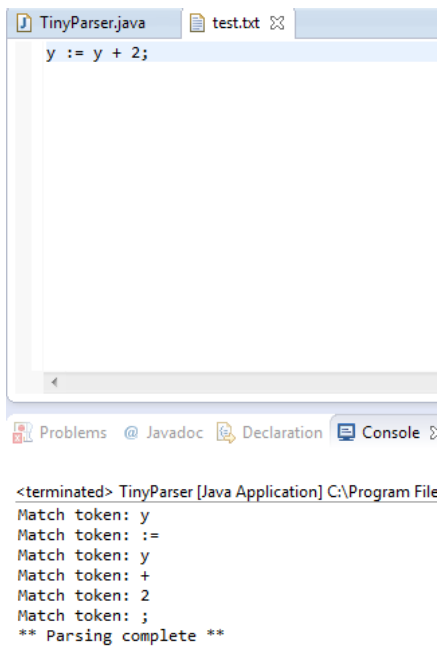
If yes, it prints “Parsing complete”.

If not, it prints “Parsing error” with hints about the error place of the received token.

It also prints the matched token values.

## Testing screenshots:

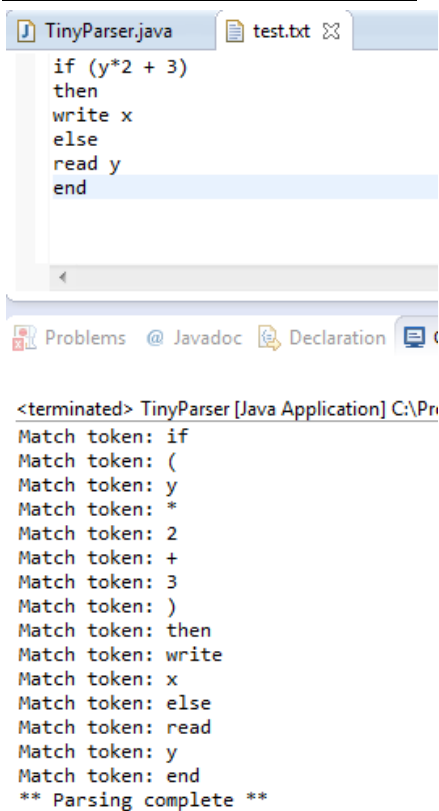
### assignment statement:



The screenshot shows an IDE with two tabs: 'TinyParser.java' and 'test.txt'. The 'test.txt' tab is active and contains the code 'y := y + 2;'. Below the code editor, there is a console window showing the output of the parser. The output indicates that the parser successfully matched all tokens and completed parsing.

```
<terminated> TinyParser [Java Application] C:\Program File
Match token: y
Match token: :=
Match token: y
Match token: +
Match token: 2
Match token: ;
** Parsing complete **
```

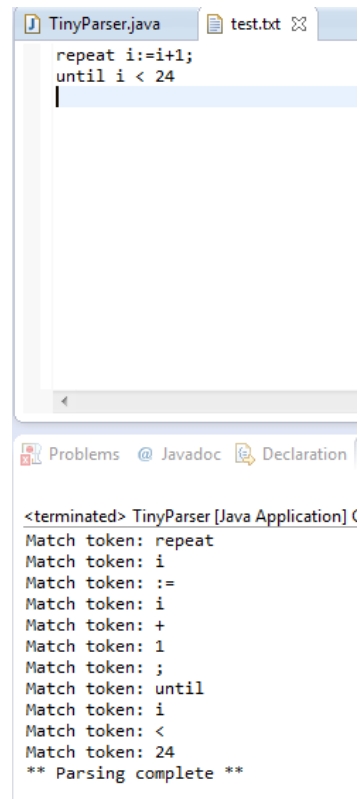
### if statement – read – write:



The screenshot shows an IDE with two tabs: 'TinyParser.java' and 'test.txt'. The 'test.txt' tab is active and contains the code 'if (y\*2 + 3) then write x else read y end'. Below the code editor, there is a console window showing the output of the parser. The output indicates that the parser successfully matched all tokens and completed parsing.

```
<terminated> TinyParser [Java Application] C:\Pr
Match token: if
Match token: (
Match token: y
Match token: *
Match token: 2
Match token: +
Match token: 3
Match token: )
Match token: then
Match token: write
Match token: x
Match token: else
Match token: read
Match token: y
Match token: end
** Parsing complete **
```

### repeat:



The screenshot shows an IDE with two tabs: 'TinyParser.java' and 'test.txt'. The 'test.txt' tab is active and contains the code 'repeat i:=i+1; until i < 24'. Below the code editor, there is a console window showing the output of the parser. The output indicates that the parser successfully matched all tokens and completed parsing.

```
<terminated> TinyParser [Java Application] C
Match token: repeat
Match token: i
Match token: :=
Match token: i
Match token: +
Match token: 1
Match token: ;
Match token: until
Match token: i
Match token: <
Match token: 24
** Parsing complete **
```