

# Lab 01

CSE2024: Programming Language Concept

---

# 과제 01) 수식 계산기 (Calc.java)

- 재귀-하강 파서 / 계산기 확장 구현 (Java)

- 문법 (EBNF)

$\langle \text{expr} \rangle \rightarrow \langle \text{bexp} \rangle \{ \& \langle \text{bexp} \rangle \mid ' \mid \langle \text{bexp} \rangle \} \mid !\langle \text{expr} \rangle \mid \text{true} \mid \text{false}$

$\langle \text{bexp} \rangle \rightarrow \langle \text{aexp} \rangle [\langle \text{relop} \rangle \langle \text{aexp} \rangle]$

$\langle \text{relop} \rangle \rightarrow == \mid != \mid < \mid > \mid <= \mid >=$

$\langle \text{aexp} \rangle \rightarrow \langle \text{term} \rangle \{ + \langle \text{term} \rangle \mid - \langle \text{term} \rangle \}$

$\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle \{ * \langle \text{factor} \rangle \mid / \langle \text{factor} \rangle \}$

$\langle \text{factor} \rangle \rightarrow [-] ( \langle \text{number} \rangle \mid (\langle \text{aexp} \rangle) )$

$\langle \text{number} \rangle \rightarrow \langle \text{digit} \rangle \{ \langle \text{digit} \rangle \}$

- 확장

① 뺄셈 (-), 나눗셈 (/) 추가

② 비교연산 (==, !=, >, <, !) 추가

③ 논리연산 (&, |, !) 추가

# 과제 01) 수식 계산기 (Calc.java)

## ■ 재귀-하강 파서 / 계산기 확장 구현 (Java)

### • 예제 및 결과

- ①  $150+10-28+4$
- ②  $23+34/5$
- ③  $12*34+24/2+150$
- ④  $12*(34+24)/2+150$
- ⑤  $-9-17$
- ⑥  $17-7==19-6$
- ⑦  $21-9!=9+3$
- ⑧  $21/3>18*5$
- ⑨  $35/7<=60/12$
- ⑩  $!20<2*12$
- ⑪  $!-16==-(10+2*3)$
- ⑫  $!!! (2+3)*4>40$
- ⑬  $3>100\&2==2$
- ⑭  $2>3|24>=24$
- ⑮  $24<40\&36>=36\&44-4*1>20$

입력 →  
출력 →

```
Problems Javadoc Declaration Console X
Calc [Java Application] C:\Users\WSOOKMYUNG\p2\poo
>> 150+10-28+4
The result is: 136
>> 23+34/5
The result is: 29
>> 12*34+24/2+150
The result is: 570
>> 12*(34+24)/2+150
The result is: 498
>> -9-17
The result is: -26
>> 17-7==19-6
The result is: false
>> 21-9!=9+3
The result is: false
>> 21/3>18*5
The result is: false
>> 35/7<=60/12
The result is: true
>> !20<2*12
The result is: false
>> !-16==-(10+2*3)
The result is: false
>> !!!(2+3)*4>40
The result is: true
>> 3>100&2==2
The result is: false
>> 2>3|24>=24
The result is: true
>> 24<40&36>=36&44-4*1>20
The result is: true
```

# 과제 01) 수식 계산기 (Calc.java)

- 재귀-하강 파서 / 계산기 확장 구현 (Java)

- 팁

```
void parse( ) {
    token = getToken(); // get the first token
    command();           // call the parsing command
}

public static void main(String args[]) {
    Calc calc = new Calc(new PushbackInputStream(System.in));
    while(true) {
        System.out.print(">> ");
        calc.parse();
    }
}
```

```
void command( ) {
    /* command -> expr '\n' */
    int result = expr();
    if (token == '\n') /* end the parse and print the result */
        System.out.println(result);
    else error();
}
```

```
int getToken( ) { /* tokens are characters */
    while(true) {
        try {
            ch = input.read();
            if (ch == ' ' || ch == '\t' || ch == '\r') ;
            else
                if (Character.isDigit(ch)) {
                    value = number( );
                    input.unread(ch);
                    return NUMBER;
                }
            else return ch;
        } catch (IOException e) {
            System.err.println(e);
        }
    }
}

private int number( ) {
    /* number -> digit { digit } */
    int result = ch - '0';
    try {
        ch = input.read();
        while (Character.isDigit(ch)) {
            result = 10 * result + ch - '0';
            ch = input.read();
        }
    } catch (IOException e) {
        System.err.println(e);
    }
    return result;
}
```

# 과제 01) 수식 계산기 (Calc.java)

## ■ 재귀-하강 파서 / 계산기 확장 구현 (Java)

- 팁

<expr> → <bexp> {& <bexp> | '|' <bexp>} | !<expr> | true | false

<bexp> → <aexp> [<relop> <aexp>]

<relop> → == | != | < | > | <= | >=

<aexp> → <term> {+ <term> | - <term>}

<term> → <factor> {\* <factor> | / <factor>}

<factor> → [-] ( <number> | (<aexp>) )

<number> → <digit> {<digit>}

```
int expr( ) {  
    /* expr -> term { '+' term } */  
    int result = term();  
    while (token == '+') {  
        match('+');  
        result += term();  
    }  
    return result;  
}  
  
int term( ) {  
    /* term -> factor { '*' factor } */  
    int result = factor();  
    while (token == '*') {  
        match('*');  
        result *= factor();  
    }  
    return result;  
}  
  
int factor() {  
    /* factor -> '(' expr ')' | number */  
    int result = 0;  
    if (token == '(') {  
        match('(');  
        result = expr();  
        match(')');  
    }  
    else if (token == NUMBER) {  
        result = value;  
        match(NUMBER); //token = getToken();  
    }  
    return result;  
}
```

# 과제 01) 수식 계산기 (Calc.java)

- 재귀-하강 파서 / 계산기 확장 구현 (Java)
  - 팁

<expr> → <bexp> {& <bexp> | '|' <bexp>} | !<expr> | true | false

<bexp> → <aexp> [<relop> <aexp>]

<relop> → == | != | < | > | <= | >=

<aexp> → <term> {+ <term> | - <term>}

<term> → <factor> {\* <factor> | / <factor>}

<factor> → [-] ( <number> | (<aexp>) )

<number> → <digit> {<digit>}

```
Object expr() { return result; }  
Object bexp() { return result; }  
String relop() { return result; }  
int aexp() { return result; }  
int term() { return result; }  
int factor() { return result; }  
int number() { return result; }
```

# 과제 01) 수식 계산기 (Calc.java)

## ■ 재귀-하강 파서 / 계산기 확장 구현 (Java)

- 팁

<expr> → <bexp> {& <bexp> | '|' <bexp>} | !<expr> | true | false

<bexp> → <aexp> [<relop> <aexp>]

<relop> → == | != | < | > | <= | >=

<aexp> → <term> {+ <term> | - <term>}

<term> → <factor> { \* <factor> | / <factor> }

<factor> → [-] ( <number> | (<aexp>) )

<number> → <digit> {<digit>}

```
Object expr() {  
    Object result;  
    if (token == '!') {  
        match('!');  
        result = !(boolean) expr();  
    }  
    else if (token == 't') {  
        true  
    }  
    else if (token == 'f') {  
        false  
    }  
    else {  
        & <bexp> | '|' <bexp>  
    }  
    return result;  
}
```

!3+6==10