

# Lab 03

2023학년도 1학기 프로그래밍언어론

---

조교 송연주

ssyeonju@sookmyung.ac.kr

## 과제 03) 언어 S의 인터프리터 구현

- 언어 S의 인터프리터 구현 (Java)

- (1) Let문 구현을 위한 allocate 함수와 free 함수 구현

- 선언된 변수들을 위한 엔트리들을 상태 state에 추가 (allocate)

- 선언된 변수들을 위한 엔트리들을 상태 state에서 제거 (free)

- (2) 언어 S의 문법에 따라 관계 및 논리 연산 수행 기능 구현

- binaryOperation()을 확장하여 정수, 스트링 관계 연산 및 부울값의 논리연산 구현

## 과제 03) 언어 S의 인터프리터 구현

- 언어 S의 인터프리터 구현 (Java)

- 언어 S의 문법 (EBNF)

```
<stmt> → id = <expr>;  
        | '{' <stmts> '  
        | if (<expr>) then <stmt> [else <stmt>]  
        | while (<expr>) <stmt>  
        | read id;  
        | print <expr>;  
        | let <decls> in <stmts> end;
```

(3) 언어 S의 문장에 do-while문, for문을 추가하고 이를 해석하는 인터프리터 작성

```
<stmt> → ...  
        | do <stmt> while (<expr>);  
        | for (<type> id = <expr>; <expr>; id = <expr>) <stmt>
```

# 과제 03) 언어 S의 인터프리터 구현

## ■ 언어 S의 인터프리터 구현 (Java)

### • 예제 및 결과

test 폴더에 있는 예제 파일

① hi0.s

② hi2.s

③ hi3.s

④ hi4.s

⑤ hi5.s

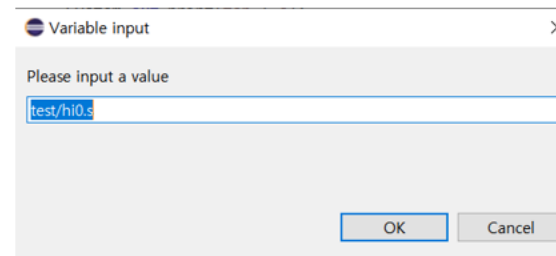
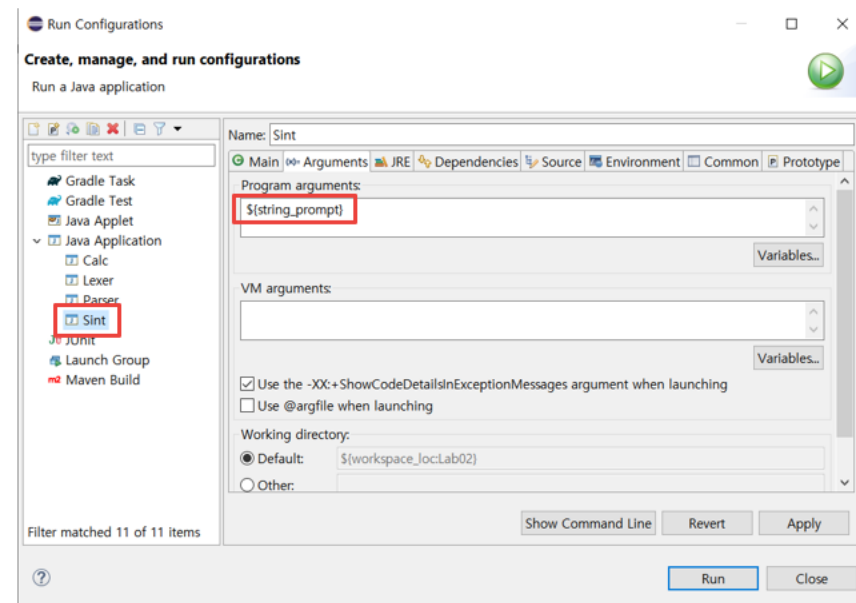
⑥ hi6.s

⑦ hi7.s

+ ⑧, ⑨ String 관계 연산 테스트 2개

+ ⑩, ⑪ 논리 연산 테스트 2개

+ ⑫ for문, ⑬ do-while문 테스트 각각 1개



# 과제 03) 언어 S의 인터프리터 구현

hi0.s

```
Begin parsing... test/hi0.s
Print
  Value: hello world!
Interpreting...test/hi0.s
hello world!

Decl
  Type: string
  Identifier: s
  Value: hello world!
Interpreting...test/hi0.s

Print
  Identifier: s
Interpreting...test/hi0.s
hello world!
```

hi2.s

```
Begin parsing... test/hi2.s
Let
  Decl
    Decl
      Type: int
      Identifier: i
    Decl
      Type: int
      Identifier: j
  Stmts
    Assignment
      Identifier: i
      Value: 1
    Print
      Value: 2^n ?
    Read
      Identifier: j
    While
      Binary
        Operator: >
        Identifier: j
        Value: 0
      Stmts
        Assignment
          Identifier: i
          Binary
            Operator: *
            Identifier: i
            Value: 2
        Assignment
          Identifier: j
          Binary
            Operator: -
            Identifier: j
            Value: 1
      Print
        Identifier: i
Interpreting...test/hi2.s
2^n ?
10
1024
```

hi3.s

```
Begin parsing... test/hi3.s
Let
  Decl
    Decl
      Type: int
      Identifier: i
      Value: 1
    Decl
      Type: int
      Identifier: sum
      Value: 0
    Decl
      Type: int
      Identifier: n
  Stmts
    Print
      Value: 1 + 2 + ... + n?
    Read
      Identifier: n
    While
      Binary
        Operator: <=
        Identifier: i
        Identifier: n
      Stmts
        Assignment
          Identifier: sum
          Binary
            Operator: +
            Identifier: sum
            Identifier: i
        Assignment
          Identifier: i
          Binary
            Operator: +
            Identifier: i
            Value: 1
      Print
        Identifier: sum
Interpreting...test/hi3.s
1 + 2 + ... + n?
10
55
```

hi4.s

```
Begin parsing... test/hi4.s
Decl
  Decl
    Type: int
    Identifier: i
    Value: 0
  Stmts
    Let
      Decl
        Decl
          Type: int
          Identifier: i
        Decl
          Type: int
          Identifier: j
      Stmts
        Assignment
          Identifier: i
          Value: 10
        Assignment
          Identifier: j
          Value: 2
        If
          Binary
            Operator: >
            Identifier: j
            Value: 0
          Assignment
            Identifier: i
            Binary
              Operator: +
              Identifier: i
              Identifier: j
          Assignment
            Identifier: i
            Binary
              Operator: -
              Identifier: i
              Identifier: j
        Print
          Identifier: i
Interpreting...test/hi4.s
12
0
```

## 과제 03) 언어 S의 인터프리터 구현

```
Begin parsing... test/hi5.s
```

```

Decl
  Decl
    Type: int
    Identifier: i
  Decl
    Type: int
    Identifier: j
  Decl
    Type: int
    Identifier: k
Stats
  Assignment
    Identifier: i
    Value: 1
  Assignment
    Identifier: j
    Value: 1
  While
    Binary
      Operator: <=
      Identifier: i
      Value: 3
    Stats
      Assignment
        Identifier: j
        Value: 1
      While
        Binary
          Operator: <=
          Identifier: j
          Value: 4
        Stats
          Assignment
            Identifier: k
            Binary
              Operator: *
              Identifier: i
              Identifier: j
          Print
            Identifier: i
          Print
            Identifier: j
          Print
            Identifier: k
          Assignment
            Identifier: j
            Binary
              Operator: +
              Identifier: j
              Value: 1
          Assignment
            Identifier: i
            Binary
              Operator: +
              Identifier: i
              Value: 1

```

## Interpreting...test/hi5.s

1  
1  
1  
2  
2  
1  
3  
3  
1  
4  
4  
2  
1  
2  
2  
2  
4  
2  
3  
6  
2  
4  
8  
3  
1  
3  
3  
2  
6  
3  
3  
9  
3  
4  
12

```
Begin parsing... test/hi6.s
```

```

let
  Decls
    Decl
      Type: int
      Identifier: i
      Value: 0
    Stmts
      Let
        Decls
          Decl
            Type: int
            Identifier: i
            Value: 1
          Decl
            Type: int
            Identifier: j
            Value: 2
        Stmts
          Print
            Identifier: i
          If
            Binary
              Operator: >
              Identifier: i
              Value: 0
            Assignment
              Identifier: i
              Binary
                Operator: +
                Identifier: i
                Identifier: j
            Assignment
              Identifier: i
              Binary
                Operator: -
                Identifier: i
                Identifier: j
          Print
            Identifier: i
        Let
          Decls
            Decl
              Type: int
              Identifier: k
              Value: 3
          Stmts
            Assignment
              Identifier: i
              Identifier: k
          Print
            Identifier: i

```

Interpreting...test/hi6.s

133

```
Begin parsing... test/h17.s
```

```

Let
  Decls
    Decl
      Type: int
      Identifier: i
      Value: 0
    Stmts
      Let
        Decls
          Decl
            Type: int
            Identifier: i
            Value: 1
          Decl
            Type: int
            Identifier: j
            Value: 1
          Decl
            Type: bool
            Identifier: k
            Value: true
        Stmts
          Print
            Identifier: i
          If
            Identifier: k
            Assignment
              Identifier: i
              Binary
                Operator: +
                Identifier: i
                Identifier: j
            Assignment
              Identifier: i
              Binary
                Operator: -
                Identifier: i
                Identifier: j
          Print
            Identifier: i
        Let
          Decls
            Decl
              Type: int
              Identifier: k
              Value: 0
          Stmts
            Assignment
              Identifier: k
              Binary
                Operator: +
                Identifier: i
                Identifier: k
          Print
            Identifier: i

```

Interpreting...test/hi7.s



# 과제 03) 언어 S의 인터프리터 구현

## 관계 연산 테스트

stringrelop1.s

```
stringrelop1.s ➡ ✕
1 string i = "apple";
2 string j = "banana";
3 if (i == j)
4     then print "strings are equal";
5 else
6     print "strings are not equal";
```

Begin parsing... test/stringrelop1.s

Decl

Type: string  
Identifier: i  
Value: apple

Interpreting...test/stringrelop1.s

Decl

Type: string  
Identifier: j  
Value: banana

Interpreting...test/stringrelop1.s

If

Binary

Operator: ==  
Identifier: i  
Identifier: j

Print

Value: strings are equal

Print

Value: strings are not equal

Interpreting...test/stringrelop1.s

strings are not equal

stringrelop2.s

```
stringrelop2.s ➡ ✕
1 string i = "apple";
2 string j = "banana";
3 if (i < j)
4     then print "banana is located behind the dictionary";
5 else
6     print "apple is located behind the dictionary";
```

Begin parsing... test/stringrelop2.s

Decl

Type: string  
Identifier: i  
Value: apple

Interpreting...test/stringrelop2.s

Decl

Type: string  
Identifier: j  
Value: banana

Interpreting...test/stringrelop2.s

If

Binary

Operator: <  
Identifier: i  
Identifier: j

Print

Value: banana is located behind the dictionary

Print

Value: apple is located behind the dictionary

Interpreting...test/stringrelop2.s

banana is located behind the dictionary

# 과제 03) 언어 S의 인터프리터 구현

논리 연산 테스트

logicalop1.s

```
logicalop1.s ↗ ✕
1 bool i = true;
2 bool j = true;
3 if (i & j)
4     then print "both are true";
5 else
6     print "one or both are false";
```

Begin parsing... test/logicalop1.s

Decl

Type: bool  
Identifier: i  
Value: true

Interpreting...test/logicalop1.s

Decl

Type: bool  
Identifier: j  
Value: true

Interpreting...test/logicalop1.s

If

Binary

Operator: &  
Identifier: i  
Identifier: j

Print

Value: both are true

Print

Value: one or both are false

Interpreting...test/logicalop1.s

both are true

logicalop2.s

```
logicalop2.s ↗ ✕
1 bool i = true;
2 bool j = false;
3 if (i | j)
4     then print "one or both are true";
5 else
6     print "both are false";
```

Begin parsing... test/logicalop2.s

Decl

Type: bool  
Identifier: i  
Value: true

Interpreting...test/logicalop2.s

Decl

Type: bool  
Identifier: j  
Value: false

Interpreting...test/logicalop2.s

If

Binary

Operator: |  
Identifier: i  
Identifier: j

Print

Value: one or both are true

Print

Value: both are false

Interpreting...test/logicalop2.s

one or both are true



# 과제 03) 언어 S의 인터프리터 구현

for문, do-while문 테스트

for.s

for.s

1 for (int i=0; i<10; i = i+1) print i;

Begin parsing... test/for.s

Let

Decls

Decl

Type: int  
Identifier: i  
Value: 0

Stmts

While

Binary

Operator: <  
Identifier: i  
Value: 10

Stmts

Print

Identifier: i

Assignment

Identifier: i

Binary

Operator: +  
Identifier: i  
Value: 1

Interpreting...test/for.s

0  
1  
2  
3  
4  
5  
6  
7  
8  
9

dowhile.s

Begin parsing... test/dowhile.s

dowhile.s

1 let  
2 int i = 5;  
3 in  
4 do {  
5 print i;  
6 i = i - 1;  
7 }  
8 while (i > 0);  
9 end;

Let

Decls

Decl

Type: int  
Identifier: i  
Value: 5

Stmts

Stmts

Print

Identifier: i

Assignment

Identifier: i

Binary

Operator: -  
Identifier: i  
Value: 1

While

Binary

Operator: >  
Identifier: i  
Value: 0

Stmts

Print

Identifier: i

Assignment

Identifier: i

Binary

Operator: -  
Identifier: i  
Value: 1

Interpreting...test/dowhile.s

5  
4  
3  
2  
1

9

## 과제 03) 언어 S의 인터프리터 구현

- 언어 S의 인터프리터 구현 (Java)

- 팁 (Sint.java - Let)

```
State Eval(Let l, State state) {
    State s = allocate(l.decls, state);
    s = Eval(l.stmts, s);
    return free(l.decls, s);
}

State allocate (Decls ds, State state) {
    if (ds != null) {
        // add entries for declared variables on the state
    }
    return null;
}

State free (Decls ds, State state) {
    if (ds != null) {
        // free the entries for declared variables from the state
    }
    return null;
}
```

## 과제 03) 언어 S의 인터프리터 구현

- 언어 S의 인터프리터 구현 (Java)
  - 팁 (Sint.java – binaryOperation)

```
Value binaryOperation(Operator op, Value v1, Value v2) {
    check(!v1.undef && !v2.undef, "reference to undef value");
    switch (op.val) {
        case "+":
            return new Value(v1.intValue() + v2.intValue());
        case "-":
            return new Value(v1.intValue() - v2.intValue());
        case "*":
            return new Value(v1.intValue() * v2.intValue());
        case "/":
            return new Value(v1.intValue() / v2.intValue());

        // relational operations

        // logical operations and or not

        default:
            throw new IllegalArgumentException("no operation");
    }
}
```

## 과제 03) 언어 S의 인터프리터 구현

- 언어 S의 인터프리터 구현 (Java)

- 팁 (Parser.java – dowhile, for)

**dowhile** : 1번 이상 반복  
do <stmt> while (<expr>);  
=  
{  
  <stmt>  
  while (<expr>)  
    <stmt>  
}

cf. while : 0번 이상 반복

**for**  
for (<type> id = <expr>; <expr>; id=<expr>) <stmt>  
=  
let  
  <type> id = <expr>  
in  
  while (<expr>)  
    <stmt>  
end

## + ) 과제 제출 기한

- 과제 제출 기한
  - 01분반 : 5월 02일 화요일 오후 11:59
  - 02분반 : 5월 03일 수요일 오후 11:59
- 제출 장소
  - 스노우보드 과제 제출 페이지에 업로드
- 추가 제출
  - 제출 기한 이후 24시간 이내 조교 메일로 전송 : 10% 감점
  - 그 이후는 20% 감점

## + ) 과제 제출 양식

- 소스파일(.java)과 과제보고서(.docx)가 담기 **압축파일(.zip 등)** 제출

- 압축 파일 이름 : **Lab과제번호\_학번\_이름**

예) Lab03\_2231013\_송연주

AST.java	2022-04-20 오전 10:23	Java 원본 파일	8KB
Parser.java	2023-03-31 오후 8:56	Java 원본 파일	11KB
Sint.java	2023-04-10 오후 2:58	Java 원본 파일	9KB
프로그래밍언어론 과제보고서 양식.docx	2023-04-11 오후 5:12	Microsoft Word ...	21KB

- 소스파일 이름

- PPT에 제시

- 과제보고서 양식

- 스노우보드에서 다운로드
- 프로그램 전체 코드 및 프로그램 설명
- 실행 결과 화면을 캡처한 이미지 첨부



Lab03\_2231013\_  
송연주.zip

## + ) 과제 질문

- ssyeonju@sookmyung.ac.kr (조교 메일로 질문 보내기)
- 질문 시 주의사항

- 메일에 반드시 과목, 분반, 전공, 이름, 학번 명시
- 몇 번 과제에서 어떤 부분이 막혔는지, 어떤 과정이 문제인지 **설명 첨부**  
(그냥 코드만 보내면 어디가 문제인지 알 수 없어요)
- 답장이 늦을 수 있으니 이 점 고려하여 미리 질문 (특히 과제 제출 마지막날 유의!)
- 그 외 출석 등 다른 질문들도 메일로

## + ) 참고

- 채점시 고려하는 기본 사항
  - 보고서 구성에 맞게 보고서를 작성
    - ① 소스코드 프로그램 전체 코드
    - ② 프로그램 설명
    - ③ 실행 화면 캡처
  - 확장
    - ① 빌드(컴파일)가 가능한지 테스트
    - ② 제시된 예제가 출력되는지 테스트
    - ③ 의도에 맞게 출력되는지 테스트