

# Lab 05

2023학년도 1학기 프로그래밍언어론

---

조교 송연주

ssyeonju@sookmyung.ac.kr

## 과제 05) 언어 S의 파서, 인터프리터 확장 구현

- 언어 S의 함수 관련 기능 추가 구현 (Java)

- (1) 함수 파싱과 AST 구현

함수 정의 :  $\langle \text{function} \rangle \rightarrow \text{fun } \langle \text{type} \rangle \text{ id} ( \langle \text{params} \rangle ) \langle \text{stmt} \rangle$

$\langle \text{params} \rangle \rightarrow \langle \text{type} \rangle \text{ id } \{, \langle \text{type} \rangle \text{ id} \}$

함수 호출 :  $\langle \text{stmt} \rangle \rightarrow \text{id} ( \langle \text{expr} \rangle \{, \langle \text{expr} \rangle \} ) ;$

리턴문 :  $\langle \text{stmt} \rangle \rightarrow \text{return } \langle \text{expr} \rangle ;$

- (2) 인터프리터에 함수 구현

함수 정의

함수 호출 (반환값이 있는 경우)

함수 반환

## 과제 05) 언어 S의 파서, 인터프리터 확장 구현

- 언어 S의 함수 관련 기능 추가 구현 (Java)

- 언어 S의 함수 관련 문법 (EBNF)

<command> → <decl> | <stmt> | <function>

<function> → fun <type> id( <params> ) <stmt>

<params> → <type> id {, <type> id}

<type> → int | bool | string | void

<stmt> → ...

| return <expr> // 리턴문

| id(<expr> {, <expr>}); // 함수 호출

<factor> → ...

| id( <expr> {, <expr>}); // 함수 호출

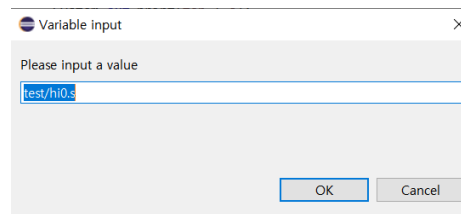
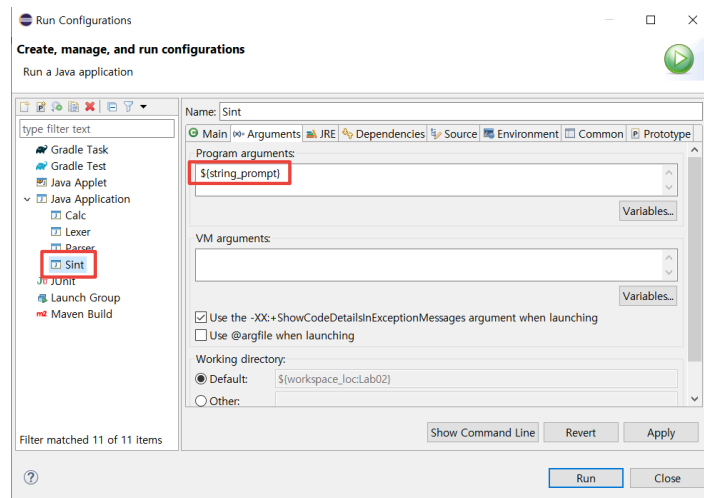
# 과제 05) 언어 S의 파서, 인터프리터 확장 구현

## ■ 언어 S에 배열 선언 및 사용기능 추가 (Java)

### • 예제 및 결과

test 폴더에 있는 예제 파일

- ① hi8.s
- ② hi9.s
- ③ hi10.s
- ④ hi11.s
- ⑤ hi12.s
- ⑥ hi13.s



## 과제 05) 언어 S의 파서, 인터프리터 확장 구현

hi8.s

```
Begin parsing... test/hi8.s
Interpreting...test/hi8.s
Interpreting...test/hi8.s
1
Interpreting...test/hi8.s
Interpreting...test/hi8.s
100
```

hi10.s

```
Begin parsing... test/hi10.s
Interpreting...test/hi10.s
Interpreting...test/hi10.s
Interpreting...test/hi10.s
Interpreting...test/hi10.s
11
11
15
15
```

hi12.s

```
Begin parsing... test/hi12.s
Interpreting...test/hi12.s
Interpreting...test/hi12.s
Interpreting...test/hi12.s
100
100
```

hi9.s

```
Begin parsing... test/hi9.s
Interpreting...test/hi9.s
Interpreting...test/hi9.s
Interpreting...test/hi9.s
Interpreting...test/hi9.s
Interpreting...test/hi9.s
Interpreting...test/hi9.s
120
Interpreting...test/hi9.s
Interpreting...test/hi9.s
true
```

hi11.s

```
Begin parsing... test/hi11.s
Interpreting...test/hi11.s
Interpreting...test/hi11.s
120
```

hi13.s

```
Begin parsing... test/hi13.s
Interpreting...test/hi13.s
120
true
```

## 과제 05) 언어 S의 파서, 인터프리터 확장 구현

- 언어 S에 함수 관련 기능 추가 구현 (Java)

- Parser.java

함수 정의

```
private Function function() {  
    // <function> -> fun <type> id(<params>) <stmt>  
    match(Token.FUN);  
    Type t = type();  
    String str = match(Token.ID);  
    funId = str;  
    Function f = new Function(str, t);  
    match(Token.LPAREN);  
    if (token != Token.RPAREN)  
        f.params = params();  
    match(Token.RPAREN);  
    Stmt s = stmt();  
    f.stmt = s;  
    return f;  
}
```

```
private Decls params() {  
    Decls params = new Decls();  
    /*  
    parse declarations of parameters  
    */  
    return params;  
}
```

## 과제 05) 언어 S의 파서, 인터프리터 확장 구현

- 언어 S에 함수 관련 기능 추가 구현 (Java)
  - Parser.java

함수 호출

```
private Call call(Identifier id) {  
    // <call> -> id(<expr> {, <expr>});  
    match(Token.LPAREN);  
    Call c = new Call(id, arguments());  
    match(Token.RPAREN);  
    match(Token.SEMICOLON);  
    return c;  
}
```

리턴문

```
private Return returnStmt() {  
    // <returnStmt> -> return <expr>;  
    match(Token.RETURN);  
    Expr e = expr();  
    match(Token.SEMICOLON);  
    return new Return(funId, e);  
}
```

## 과제 05) 언어 S의 파서, 인터프리터 확장 구현

- 언어 S에 함수 관련 기능 추가 구현 (Java)

- AST.java

함수 정의

```
class Function extends Command {  
    // Function = Type type; Identifier id; Decls params; Stmt stmt  
    Identifier id;  
    Decls params;  
    Stmt stmt;  
  
    Function(String s, Type t) {  
        id = new Identifier(s); type = t; params = null; stmt = null;  
    }  
  
    public String toString ( ) {  
        return id.toString()+params.toString();  
    }  
}
```

```
class Value extends Expr {  
    // Value = int | bool | string | array | function  
    protected boolean undef = true;  
    Object value = null; // Type type;  
  
    Value(Type t) {  
        type = t;  
        if (type == Type.INT) value = new Integer(0);  
        if (type == Type.BOOL) value = new Boolean(false);  
        if (type == Type.STRING) value = "";  
        undef = false;  
    }  
  
    Value(Object v) {  
        if (v instanceof Function) type = Type.FUN;  
        value = v; undef = false;  
    }  
  
    Function funValue ( ) {  
        if (value instanceof Function)  
            return (Function) value;  
        else return null;  
    }  
}
```



## 과제 05) 언어 S의 파서, 인터프리터 확장 구현

- 언어 S에 함수 관련 기능 추가 구현 (Java)

- AST.java

함수 호출

```
class Call extends Expr {
    Identifier fid;
    Exprs args;

    Call(Identifier id, Exprs a) {
        fid = id;
        args = a;
    }
}
```

리턴문

```
class Return extends Stmt {
    Identifier fid;
    Expr expr;

    Return (String s, Expr e) {
        fid = new Identifier(s);
        expr = e;
    }
}
```

## 과제 05) 언어 S의 파서, 인터프리터 확장 구현

- 언어 S에 함수 관련 기능 추가 구현 (Java)
  - Sint.java

함수 정의

```
State Eval(Command c, State state) {  
    if (c instanceof Decl) {  
        Decls decls = new Decls();  
        decls.add((Decl) c);  
        return allocate(decls, state);  
    }  
  
    if (c instanceof Function) {  
        Function f = (Function) c;  
        state.push(f.id, new Value(f));  
        return state;  
    }  
}
```

## 과제 05) 언어 S의 파서, 인터프리터 확장 구현

- 언어 S에 함수 관련 기능 추가 구현 (Java)

- Sint.java

함수 호출

반환값이 있는 함수 호출

```
Value V(Expr e, State state) {  
  
    if (e instanceof Call)  
        return V((Call)e, state);  
    throw new IllegalArgumentException("no operation");  
}
```

반환값이 없는 함수 호출

```
State Eval Stmt s, State state {  
    if (s instanceof Call)  
        return Eval((Call)s, state);  
    if (s instanceof Return)  
        return Eval((Return)s, state);  
    throw new IllegalArgumentException("no statement");  
}
```

```
// value-returning call  
Value V(Call c, State state) {  
    Value v = state.get(c.fid);  
    Function f = v.funValue();  
    State s = newFrame(state, c, f);  
    s = Eval(f.stmt, s);  
    v = s.peek().val;  
    s = deleteFrame(s, c, f);  
    return v;  
}  
  
// call without return value  
State Eval(Call c, State state) {  
    // evaluate call without return value  
    return null;  
}
```

## 과제 04) 언어 S의 파서, 인터프리터 확장 구현

### ■ 언어 S에 함수 관련 기능 추가 구현 (Java)

- Sint.java

프레임 구성과 매개변수 전달

1. 인자 값 계산
2. 형식 매개변수들을 위한 기억공간 할당
3. 인자 값들을 형식 매개변수에 복사
4. 프레임에 반환 값 엔트리를 매개변수 바로 위에 추가



함수 반환

```
State Eval(Return r, State state) {  
    Value v = V(r.expr, state);  
    return state.set(new Identifier("return"), v);  
}
```

```
State newFrame (State state, Call c, Function f) {  
    if (c.args.size() == 0)  
        return state;  
    Value val[] = new Value[f.params.size()];  
    int i = 0;  
    // 인자 값을 계산하여 그 값을 val[]에 저장  
    for (Expr e: c.args)  
        val[i++] = V(e, state);  
    // 현재 상태에 매개변수 기억공간 할당(allocate 사용)  
    // 인자의 값을 매개변수에 전달  
    // 프레임에 반환 값을 위한 엔트리 추가  
    // 상태 반환  
    return null;  
}
```

```
State deleteFrame (State state, Call c, Function f) {  
    // 프레임에서 반환 값 엔트리 제거  
    // 프레임에서 매개변수를 위한 기억공간 제거(free 사용)  
    return null;  
}
```

## + ) 과제 제출 기한





- 과제 제출 기한
  - 01분반 : 5월 30일 화요일 오후 11:59
  - 02분반 : 5월 31일 수요일 오후 11:59
- 제출 장소
  - 스노우보드 과제 제출 페이지에 업로드
- 추가 제출
  - 제출 기한 이후 24시간 이내 조교 메일로 전송 : 10% 감점
  - 그 이후 : 20% 감점

## + ) 과제 제출 양식

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

- 압축 파일 이름 : **Lab과제번호\_학번\_이름**

예) Lab05\_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



## + ) 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