

컴파일러개론

-4주차-

분반: 00

학번: 201404377

이름: 진승언

[문제 해결 방법 및 테스트 실행결과]

이번 과제는 JVM 어셈블리어 코드를 해석하고 jasmin을 이용해서 class 파일을 생성하고 java코드로 변환해보고 실행결과를 확인해보는 것이다.

(1) JVM 어셈블리를 보기에 앞서 Test.java로 변환하면 다음과 같다.

```
*Test.java
1 import java.io.PrintStream;
2
3 public class Test {
4
5     public Test() {
6         // TODO Auto-generated constructor stub
7     }
8
9     public static int add(int n, int n2) {
10         int n3 = n + n2;
11         return n3;
12     }
13
14     public static void main(String[] args) {
15         int n = 33;
16         System.out.println(Test.add(1, n));
17     }
18 }
```

JVM 어셈블리코드를 살펴보면 다음과 같다. (Test.j 파일)

.class public Test

=> Test클래스이다.

```
public class Test
```

.super java/lang/Object

=> 위의 Test 클래스는 java/lang/Object를 상속받는다.

; standard initializer

=> 선언문이다.

```

5 public Test() {
6     // TODO Auto-generated constructor stub
7 }

```

.method public <init>()V

=> return type이 void인 메소드이다. (생성자이다. 기본생성자는 생략 가능)

aload_0

=> 지역변수 배열의 0번째 있는 값을 피연산자 스택으로 로드한다. 메소드의 첫 번째 참조 인수를 읽어서 스택에 넣는 것이다.

(Load reference from local variable)

invokenonvirtual java/lang/Object/<init>()V

=> invokenonvirtual 명령어는 인스턴스 초기화 메소드와 현재 클래스의 super class private 메소드 및 메소드를 호출하는 데 사용된다.

.end method

=> 위의 생성자 메소드 끝

```

9 public static int add(int n, int n2) {
10     int n3 = n + n2;
11     return n3;
12 }
13

```

.method public static add(II)I

=> return type이 int 형이고 매개변수로 int형 두 개를 받는 메소드

.limit stack 32

.limit locals 32

=> 스택 32 크기만큼 만듦

iload_0

iload_1

=> 0, 1 (실제론 주소) 공간을 불러온다. (매개변수로 받은 n과 n2를 불러오는

거다)

iadd

=> int형을 더한다. operand 스택에서 pop된다. (스택의 n 과 n2 값이 pop 되어 n + n2 더한다)

istore_2

=> 지역변수에 int형 값을 저장한다.

iload_2

=> int형 지역 변수를 로드한다. (n3에 저장함)

ireturn

=> 메소드 안에서 int형 값을 리턴한다.

.end method

=> add 메소드 끝

```
13  
14 public static void main(String[] args) {  
15     int n = 33;  
16     System.out.println(Test.add(1, n));  
17 }  
18
```

.method public static main([Ljava/lang/String;)V

=> void type의 static 메소드를 만든다.

.limit stack 32

.limit locals 32

=> 스택 32 크기만큼 만듦

ldc 33

=> 상수 33을 ldc에 저장

istore_2

=> run-time constant pool로부터 아이템을 push한다. 상수 33을 ldc에 저장하고 ldc를 2라는 공간에 저장한다. (int n = 33)

getstatic java/lang/System/out Ljava/io/PrintStream;

=> 다른 클래스에 정의된 static 메소드인 PrintStream을 호출한다.

ldc 1

=> 1을 ldc에 저장

iload_2

=> 변수에 앞서 저장한 1을 저장 add()의 1을 의미한다.

invokestatic Test/add(II)I

=> 현재 클래스에 정의된 클래스(정적) 메서드 호출. 즉 Test 클래스의 return type이 int형이고 매개변수는 int형 2개를 받는 add라는 함수를 호출한다. Test.add();를 의미한다.

invokevirtual java/io/PrintStream/println(I)V

=> 현재 클래스에 정의된 클래스(정적) 메서드인 println()을 호출한다.

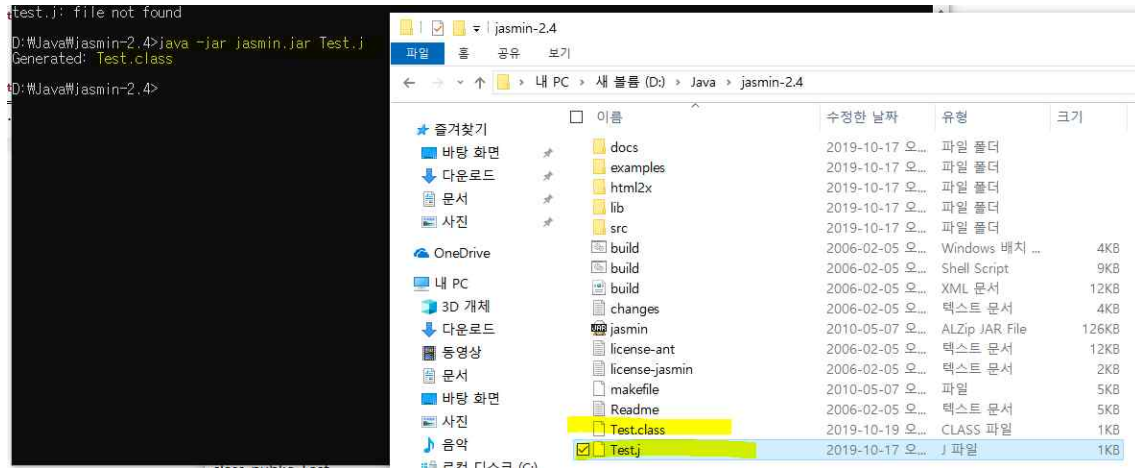
return

=> 리턴한다. (리턴하는 값은 없다.)

.end method

=> main 메소드 끝

과제 (2)의 jasmin을 설치 후 앞서 작성한 Test.j를 `java -jar jasmin.jar Test.j` 명령을 통해 class파일을 생성한 후



(3)의 `java Test` 명령어를 통해 `Test.class`를 실행하면 다음과 같이 34라는 결과가 나온다.

