

컴파일러개론

-hw 05-

분반: 00

학번: 201404377

이름: 진승언

문제 해결 방법

먼저 이번 과제의 java코드는 다음과 같이 sum(int n)이라는 메소드를 만들어 반복문을 사용하여 0부터 n까지 더한 결과값을 리턴한다. main에서는 이 메소드를 호출하고 리턴 받은 값을 출력하게 구현했다.

```
1  /*
2  .class public JvmTest
3  .super java/lang/Object
4  ; standard initializer
5  */
6  public class JvmTest {
7
8      /*
9      .method public <init>()V
10     aload_0
11     invokevirtual java/lang/Object/<init>()V
12     return
13     .end method
14     */
15     public JvmTest() {
16     }
17
18     /*
19     .method public static sum(I)I
20     .limit stack 32
21     .limit locals 32
22     iconst_0
23     istore_1
24     iconst_1
25     istore_2
26     goto 22
27     17: iload_1
28     iload_2
29     iadd
30     istore_1
31     iinc           2 1
32     22: iload_2
33     iload_0
34     if_icmple     17
35     iload_1
36     ireturn
37     .end method
38     */
39     public static int sum(int n) {
40         int sum = 0;
41         for (int i = 1; i <= n; i++) {
42             sum += i;
43         }
44         return sum;
45     }
46
47     /*
48     .method public static main([Ljava/lang/String;)V
49     .limit stack 32
50     .limit locals 32
51     getstatic java/lang/System/out Ljava/io/PrintStream;
52     bipush      100
53     invokestatic JvmTest/sum(I)I
54     invokevirtual java/io/PrintStream/println(I)V
55     return
56     .end method
57     */
58
59     public static void main(String[] args) {
60         System.out.println(sum(100));
61     }
62 }
63
```

위 코드의 주석에 나와있는 것처럼 이 java코드를 JVM 어셈블리로 다음과 같이 구현했다.

```

라인
1  .class public JvmTest
2  .super java/lang/Object
3  ; standard initializer
4  .method public <init>() V
5      aload_0
6      invokevirtual java/lang/Object/<init>() V
7      return
8  .end method
9  .method public static sum(I) I
10     .limit stack 32
11     .limit locals 32
12     iconst_0
13     istore_1
14     iconst_1
15     istore_2
16     goto 22
17     ; iload_1
18     iload_2
19     iadd
20     istore_1
21     inc 2 1
22     iload_2
23     iload_0
24     if_icmple 17
25     iload_1
26     ireturn
27 .end method

28 .method public static main([Ljava/lang/String;) V
29     .limit stack 32
30     .limit locals 32
31     gets static java/lang/System/out Ljava/io/PrintStream;
32     bipush 100
33     invokestatic JvmTest/sum(I)I
34     invokevirtual java/io/PrintStream/println(I) V
35     return
36 .end method

```

1-3라인: 클래스를 만든다.

4-8라인: 생성자이다. 반환형이 V(void)이고 생성자이므로 <init>이 호출된다.

9-27라인: public static int sum(int n) 구현부이다. stack과 locales를 공간이 충분하게 32로 해놨다. iconst_0와 istore_1은 0의 값을 가진 변수를 저장소 1에 저장한다. 즉 `int sum = 0;`을 의미한다. iconst_1과 istore_2는 for문안의 `int i = 1`을 의미한다. 그리고 비교 및 반복문은 if_icmple과 goto를 통해 구현했다. 그리고 1씩 증가하는 연산자는 iadd로 나타내고 반환은 ireturn을 통해 한다. Locale을 표로 나타내면 다음과 같을 것이다.

0	1	2	3	4
int n	int sum	int i		

29-37라인: `public static void main(String[] args)` 을 구현한거다. `invokestatic JvmTest/sum(I)I` ,`invokevirtual java/io/PrintStream/println(I)V` 를 통해 앞서 구현한 `sum()` 메소드를 호출한다. 그리고 출력한다.

테스트 코드 실행 결과(캡처)

위의 `JvmTest.j`를 `Jasmin`으로 `class` 파일로 만들고 실행한 결과이다.
(1에서 100까지의 더한 값을 출력하는 예제이다.)

```
82107@DESKTOP-3D28IO3 MINGW64 /d/Java/jasmin-2.4
$ java -jar jasmin.jar JvmTest.j
Generated: JvmTest.class

82107@DESKTOP-3D28IO3 MINGW64 /d/Java/jasmin-2.4
$ java JvmTest
5050

82107@DESKTOP-3D28IO3 MINGW64 /d/Java/jasmin-2.4
```

`class`파일이 생성되고 실행 시 5050이라고 결과값이 맞게 출력됨을 볼 수 있다.

총 과제 수행에 걸린 시간

4시간