



# 컴파일러개론

과제 9 주차

안상준 | 202002514 | 2024.11.08

## 과제 2

```
1 // Source code is decompiled from a .class file using FernFlower decompiler.
2 public class Test {
3     public Test() {
4     }
5
6     public static int cal(int var0, int var1) {
7         return (var0 * var1 - 2) / 3;
8     }
9
10    public static void main(String[] var0) {
11        byte var2 = 13;
12        System.out.println(cal(2, var2));
13    }
14 }
15
```

Test.j 파일을 jasmin 을 이용하여 class 파일로 만든 결과이다.

```

sangjun19 ~ /Desktop/Compiler/실습/9/src main ▸ javap -v Test.class
Classfile /Users/sangjun/Desktop/Compiler/실 습 /9/src/Test.class
  Last modified 2024. 11. 8.; size 463 bytes
  SHA-256 checksum 4fa7c97c702a14545e2f9df4a6011268113a282b2bd74e1f068bf3c2b5c65b04
  Compiled from "Test.java"
public class Test
  minor version: 0
  major version: 65
  flags: (0x0021) ACC_PUBLIC, ACC_SUPER
  this_class: #14                // Test
  super_class: #2                // java/lang/Object
  interfaces: 0, fields: 0, methods: 3, attributes: 1
Constant pool:
  #1 = Methodref                #2.#3      // java/lang/Object."<init>":()V
  #2 = Class                    #4         // java/lang/Object
  #3 = NameAndType              #5:#6      // "<init>":()V
  #4 = Utf8                    java/lang/Object
  #5 = Utf8                    <init>
  #6 = Utf8                    ()V
  #7 = Fieldref                #8.#9      // java/lang/System.out:Ljava/io/PrintStream;
  #8 = Class                    #10        // java/lang/System
  #9 = NameAndType              #11:#12    // out:Ljava/io/PrintStream;
  #10 = Utf8                   java/lang/System
  #11 = Utf8                   out
  #12 = Utf8                   Ljava/io/PrintStream;
  #13 = Methodref              #14.#15    // Test.cal:(II)I
  #14 = Class                  #16        // Test
  #15 = NameAndType            #17:#18    // cal:(II)I
  #16 = Utf8                   Test
  #17 = Utf8                   cal
  #18 = Utf8                   (II)I
  #19 = Methodref              #20.#21    // java/io/PrintStream.println:(I)V
  #20 = Class                  #22        // java/io/PrintStream
  #21 = NameAndType            #23:#24    // println:(I)V
  #22 = Utf8                   java/io/PrintStream
  #23 = Utf8                   println
  #24 = Utf8                   (I)V
  #25 = Utf8                   Code
  #26 = Utf8                   LineNumberTable
  #27 = Utf8                   main
  #28 = Utf8                   ([Ljava/lang/String;)V
  #29 = Utf8                   SourceFile
  #30 = Utf8                   Test.java
{
  public Test();
    descriptor: ()V
    flags: (0x0001) ACC_PUBLIC
    Code:
      stack=1, locals=1, args_size=1
         0: aload_0
         1: invokespecial #1                // Method java/lang/Object."<init>":()V
         4: return
    LineNumberTable:
      line 4: 0
      line 6: 4

```

```

line 8: 4
public static int cal(int, int);
descriptor: (II)I
flags: (0x0009) ACC_PUBLIC, ACC_STATIC
Code:
    stack=2, locals=2, args_size=2
        0: iload_0
        1: iload_1
        2: imul
        3: iconst_2
        4: isub
        5: iconst_3
        6: idiv
        7: ireturn
LineNumberTable:
    line 9: 0

public static void main(java.lang.String[]);
descriptor: ([Ljava/lang/String;)V
flags: (0x0009) ACC_PUBLIC, ACC_STATIC
Code:
    stack=3, locals=2, args_size=1
        0: bipush      13
        2: istore_1
        3: getstatic    #7                // Field java/lang/System.out:Ljava/io/PrintStream;
        6: iconst_2
        7: iload_1
        8: invokestatic #13                // Method cal:(II)I
       11: invokevirtual #19              // Method java/io/PrintStream.println:(I)V
       14: return
LineNumberTable:
    line 13: 0
    line 14: 3
    line 15: 14
}
SourceFile: "Test.java"

```

aload\_0 은 this 를 의미한다.

cal 함수에서 (II)I 는 정수타입 매개변수 두개를 인자로 취하며 반환은 정수타입임을 의미한다. iload\_0, iload\_1 은 로컬변수 슬롯 0, 1 에서 값을 가져와 스택에 push 하게 되는데 각 로컬변수 슬롯에는 매개변수로 값이 각 들어가게 된다. imul 명령어를 통해 스택에서 두 개의 값을 pop 하여 결과값을 다시 push 한다. iconst\_2 는 상수 2 를 스택에 push 하고 isub 을 통해 다시 stack 에서 값을 가져와 뺄셈하여 결과값을 새택에 push 한다 iconst\_3 는 상수를 스택에 넣고 idiv 는 스택에서 값을 pop 하여 나누어 결과를 push 한다 ireturn 은 스택에 있는 값을 정수형태로 반환한다.

다음은 main 함수다. 함수 선언후 bipush 13 을 하게 되는데 Test.j 에는 ldc13 으로 돼있어 차이가 있다. ldc13 은 값의 범위가 0~65535 되며, bipush 의 경우 - 128~127 이다. bipush 13, ldc13 모두 stack 의 13 이라는 상수값을 push 하게 된다. istore\_1 은 로컬변수슬롯 1 에 스택의 값을 가져와 저장한다. getstatic 은 출력함수 객체를 생성한다. icondt\_2 는 상수값 2 를 스택에 push 하고 iload\_1 을 통해 로컬변수 슬롯 1 에 있는 값을 스택에 push 한다. 즉 13 이 스택에 push 된다. invokestatic // Method cal:(II)I 를 통해 cal 함수를 호출한다. 이때 스택에 있는 값 2, 13 이 매개변수가

되게 된다. invokevirtual // Method java/io/PrintStream.println(I)V 로 cal 함수의 반환값이 출력되게 된다.