

컴파일러 10주차 실습 보고서

201702086 한정경

```
han@hanjeong-gyeong-ui-MacBookAir jasmin-2.4 % java -jar jasmin.jar Test.j
Generated: Test.class
```

그림 1 jasmin 명령어 사용

jasmin-2.4 폴더 경로에 명시된 깃허브의 Test.j를 다운로드한다. java -jar jasmin.jar [name].j 명령어를 통해 Test.j를 .class로 컴파일한다. 이후 java [name] 명령어를 통해 .class 파일을 실행한다.

```
han@hanjeong-gyeong-ui-MacBookAir jasmin-2.4 % java Test
34
```

그림 2 Test.class 실행 결과

class 파일을 java로 실행한 결과 34가 나오는 것을 확인할 수 있다.

```
1 public class Main {
2     public static void main(String[] args)
3     {
4         int num1 = 5;
5         int num2 = 10;
6         int num3 = num1 + num2;
7         System.out.println(num3);
8         System.out.println("Hello world!");
9     }
10 }
```

그림 3 수기로 작성한 java code

수기로 작성한 Main.java에 대해 javac Main.java 명령어를 사용하여 Main.class를 생성한다.

```
han@hanjeong-gyeong-ui-MacBookAir jasmin-2.4 % java Main
15
Hello world!
```

그림 4 Main.class 실행 결과

그림 3은 직접 작성한, Hello world! 를 출력하는 java코드로부터 컴파일한 class 대한 실행 결과이다. 올바른 결과가 나오는 것을 확인할 수 있다.

javap를 사용할 때 추가 정보를 확인하기 위해 옵션을 사용한다. 스택 사이즈, 변수 개수 등의 값을 확인하기 위한 -v 옵션. 줄과 지역변수 테이블 출력을 위한 -l 옵션. 역어셈블된 코드를 표기하기 위한 -c 옵션을 추가하여 진행한다.

```
han@hanjeong-gyeong-ui-MacBookAir Desktop % javap -v -l -c Main.class
Classfile /Users/han/Desktop/Main.class
  last modified 2023. 11. 7.; size 463 bytes
  SHA-256 checksum 8f9f1986c94e73a42accab16d2357ad963ab7bcb085bfa98b7eb555e9ca1
  Compiled from "Main.java"
  public class Main
    minor version: 0
    major version: 65
    flags: (0x0021) ACC_PUBLIC, ACC_SUPER
    this_class: #24
    super_class: #2
    interfaces: 0, fields: 0, methods: 2, 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    // java/io/PrintStream.println:()V
    #14 = Class          #16        // java/io/PrintStream
    #15 = NameAndType     #17:#18    // println:()V
    #16 = Utf8           java/io/PrintStream
    #17 = Utf8           println
    #18 = Utf8           ()V
    #19 = String         #20        // Hello world!
    #20 = Utf8           Hello world!
    #21 = Methodref      #22.#23    // java/io/PrintStream.println:(Ljava/lang/String;)V
    #22 = Class          #24        // java/lang/String
    #23 = Utf8           println
    #24 = Class          #25        // Main
    #25 = Utf8           Main
    Code
    LineNumberTable
    #26 = Utf8           main
    #27 = Utf8           ([Ljava/lang/String;)V
    #28 = Utf8           SourceFile
    #29 = Utf8           Main.java
    #30 = Utf8           Main.java
```

그림 5 javap 명령어 사용-1

그림 3과 비교하였을 때, #16에서 num3의 출력. #20에서 Hello world!의 출력이 이뤄지는 것을 확인할 수 있다.

```
public Main();
  descriptor: ()V
  flags: (0x0001) ACC_PUBLIC
  Code:
    stack=1, locals=1, args_size=1
    0: aload_0
    1: invokespecial #1
    4: return
  LineNumberTable:
    line 1: 0

public static void main(java.lang.String[]);
  descriptor: ([Ljava/lang/String;)V
  flags: (0x0009) ACC_PUBLIC, ACC_STATIC
  Code:
    stack=2, locals=4, args_size=1
    0: iconst_5
    1: istore_1
    2: bipush    10
    4: istore_2
    5: iload_1
    6: iload_2
    7: iadd
    8: istore_3
    9: getstatic #7
    12: iload_3
    13: invokevirtual #13
    16: getstatic #7
    19: ldc       #19
    21: invokevirtual #21
    24: return
  LineNumberTable:
    line 4: 0
    line 5: 2
    line 6: 5
    line 7: 9
    line 8: 16
    line 9: 24
```

그림 6 javap 명령어 사용-2

그림 6에서 스택 사이즈, 메소드의 변수개수와 변수값, 줄과 지역변수의 테이블, 역어셈블된 코드를 확인할 수 있다. 그림 3과 비교하였을 때, 1:과 4:에서 각각 5와 10이 저장된다. num3에 num1, num2의 연산 결과를 저장하기 위해 5:, 6:에서 값을 불러오고 7:에서 더하기 연산을 수행한다. 이후 8:에서 num3에 그 값을 store한다.