컴파일러개론 W10 실습 과제 보고서

컴퓨터융합학부 202102690 임세빈

Ⅰ. Javap, jasmin 사용해보기

1. Javap

```
PS C:\Users\jij09\IdeaProjects\COMPILERINTRODUCTION\W10\src> javap -c Example.class
Compiled from "Example.java"
public class Example {
  int number;

  public Example(int);
    Code:
       0: aload_0
       1: invokespecial #1                  // Method java/lang/Object."<init>":()V
       4: aload_0
       5: iload_1
       6: putfield      #7                  // Field number:I
       9: return

  void printNumber();
    Code:
       0: getstatic     #13                 // Field java/lang/System.out:Ljava/io/PrintStream;
       3: aload_0
       4: getfield      #7                  // Field number:I
       7: invokedynamic #19,  0             // InvokeDynamic #0:makeConcatWithConstants:(I)Ljava/lang/String;
      12: invokevirtual #23                 // Method java/io/PrintStream.println:(Ljava/lang/String;)V
      15: return

  public static void main(java.lang.String[]);
    Code:
       3: dup
       4: bipush        12
       6: invokespecial #29                 // Method "<init>":(I)V
       9: astore_1
      10: aload_1
      11: invokevirtual #32                 // Method printNumber:()V
      14: return
}
```

Pdf에 주어진 Example.java 코드의 javap -c 실행결과이다.

```
PS C:\Users\jij09\IdeaProjects\COMPILERINTRODUCTION\W10\src> javap -v Example.class
Classfile /C:/Users/jij09/IdeaProjects/COMPILERINTRODUCTION/W10/src/Example.class
  Last modified 2023. 11. 9.; size 992 bytes
  SHA-256 checksum f5ac88bd5446ffe3faa118ab61eb830a7c4ddc00ffc531c799440a16a0b3b75f
  Compiled from "Example.java"
public class Example
  minor version: 0
  major version: 61
  flags: (0x0021) ACC_PUBLIC, ACC_SUPER
  this_class: #8                          // Example
  super_class: #2                         // java/lang/Object
  interfaces: 0, fields: 1, methods: 3, attributes: 3
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          // Example.number:I
   #8 = Class              #10            // Example
   #9 = NameAndType        #11:#12        // number:I
  #10 = Utf8               Example
  #11 = Utf8               number
  #12 = Utf8               I
```

```
  #13 = Fieldref           #14.#15        // java/lang/System.out:Ljava/io/PrintStream;
  #14 = Class              #16            // java/lang/System
  #15 = NameAndType        #17:#18        // out:Ljava/io/PrintStream;
  #16 = Utf8               java/lang/System
  #17 = Utf8               out
  #18 = Utf8               Ljava/io/PrintStream;
  #19 = InvokeDynamic      #0:#20         // #0:makeConcatWithConstants:(I)Ljava/lang/String;
  #20 = NameAndType        #21:#22        // makeConcatWithConstants:(I)Ljava/lang/String;
  #21 = Utf8               makeConcatWithConstants
  #22 = Utf8               (I)Ljava/lang/String;
  #23 = Methodref          #24.#25        // java/io/PrintStream.println:(Ljava/lang/String;)V
  #24 = Class              #26            // java/io/PrintStream
  #25 = NameAndType        #27:#28        // println:(Ljava/lang/String;)V
  #26 = Utf8               java/io/PrintStream
  #27 = Utf8               println
  #28 = Utf8               (Ljava/lang/String;)V
  #29 = Methodref          #8.#30         // Example."<init>":(I)V
  #30 = NameAndType        #5:#31         // "<init>":(I)V
  #31 = Utf8               (I)V
  #32 = Methodref          #8.#33         // Example.printNumber:()V
  #33 = NameAndType        #34:#6         // printNumber:()V
  #34 = Utf8               printNumber
  #35 = Utf8               Code
  #36 = Utf8               LineNumberTable
  #37 = Utf8               main
  #38 = Utf8               ([Ljava/lang/String;)V
  #39 = Utf8               SourceFile
```

```
  #40 = Utf8               Example.java
  #41 = Utf8               BootstrapMethods
  #42 = MethodHandle       6:#43          // REF_invokeStatic java/lang/invoke/StringConcatFactory.makeConcatWithConstants:(Ljava/lang/invoke/MethodHandles$Lookup;Ljava/lang/Strin
;Ljava/lang/invoke/MethodType;Ljava/lang/String;[Ljava/lang/Object;)Ljava/lang/invoke/CallSite;
  #43 = Methodref          #44.#45        // java/lang/invoke/StringConcatFactory.makeConcatWithConstants:(Ljava/lang/invoke/MethodHandles$Lookup;Ljava/lang/String;Ljava/lang/invo
e/MethodType;Ljava/lang/String;[Ljava/lang/Object;)Ljava/lang/invoke/CallSite;
  #44 = Class              #46            // java/lang/invoke/StringConcatFactory
  #45 = NameAndType        #21:#47        // makeConcatWithConstants:(Ljava/lang/invoke/MethodHandles$Lookup;Ljava/lang/String;Ljava/lang/invoke/MethodType;Ljava/lang/String;[Ljav
/lang/Object;)Ljava/lang/invoke/CallSite;
  #46 = Utf8               java/lang/invoke/StringConcatFactory
  #47 = Utf8               (Ljava/lang/invoke/MethodHandles$Lookup;Ljava/lang/String;Ljava/lang/invoke/MethodType;Ljava/lang/String;[Ljava/lang/Object;)Ljava/lang/invoke/CallSite;
  #48 = String             #49            // print: \u0001
  #49 = Utf8               print: \u0001
  #50 = Utf8               InnerClasses
  #51 = Class              #52            // java/lang/invoke/MethodHandles$Lookup
  #52 = Utf8               java/lang/invoke/MethodHandles$Lookup
  #53 = Class              #54            // java/lang/invoke/MethodHandles
  #54 = Utf8               java/lang/invoke/MethodHandles
  #55 = Utf8               Lookup
```

```
{
  int number;
    descriptor: I
    flags: (0x0000)

  public Example(int);
    descriptor: (I)V
    flags: (0x0001) ACC_PUBLIC
    Code:
      stack=2, locals=2, args_size=2
         0: aload_0
         1: invokespecial #1                  // Method java/lang/Object."<init>":()V
         4: aload_0
         5: iload_1
         6: putfield      #7                  // Field number:I
         9: return
      LineNumberTable:
        line 5: 0
        line 6: 4
        line 7: 9
```

```
  void printNumber();
    descriptor: ()V
    flags: (0x0000)
    Code:
      stack=2, locals=1, args_size=1
         0: getstatic     #13                 // Field java/lang/System.out:Ljava/io/PrintStream;
         3: aload_0
         4: getfield      #7                  // Field number:I
         7: invokedynamic #19,  0             // InvokeDynamic #0:makeConcatWithConstants:(I)Ljava/lang/String;
        12: invokevirtual #23                 // Method java/io/PrintStream.println:(Ljava/lang/String;)V
        15: return
      LineNumberTable:
}
      #48 print: \u0001
InnerClasses:
```

pdf에 주어진 Example.java의 javap -v 의 결과이다.

2. Jasmin

```
jij09@DESKTOP-ESOMH0M MINGW64 ~/IdeaProjects/COMPILERINTRODUCTION/jasmin-2.4 (ma
in)
$ java -jar jasmin.jar Test.j
Generated: Test.class
```

```
jij09@DESKTOP-ESOMH0M MINGW64 ~/IdeaProjects/COMPILERINTRODUCTION/jasmin-2.4 (ma
in)
$ ls
Readme.txt  build.bat  changes.txt  html2x/     license-ant.txt     src/
Test.class  build.sh*  docs/        jasmin.jar  license-jasmin.txt
Test.j      build.xml  examples/    lib/        makefile
```

Jasmin 으로 Test.j 파일을 Test.class 파일로 변환하였다. 아래 사진에서 Test.class 파일을 확인할 수 있다.

Ⅱ. 수기로 작성한 JAVA 코드를 javac로 컴파일, javap로 disassemble 해보기

1. Test.j

```
.class public Test
.super java/lang/Object
; strandard initializer
.method public <init>()V
aload_0
invokenonvirtual java/lang/Object/<init>()V
return
.end method
.method public static add(II)I
.limit stack 32
.limit locals 32
iload_0
iload_1
iadd
istore_2
iload_2
ireturn
.end method
.method public static main([Ljava/lang/String;)V
.limit stack 32
.limit locals 32
ldc 33
istore_2
getstatic java/lang/System/out Ljava/io/PrintStream;
ldc 1
iload_2
invokestatic Test/add(II)I
invokevirtual java/io/PrintStream/println(I)V
return
.end method
```

2. JAVA 코드로 직접 표현해보기

```java
public class Test extends Object {
    // Standard initializer
    no usages
    public Test() {
        super();
    }

    public static int add(int a, int b) {
        int result;
        result = a + b;
        return result;
    }

    public static void main(String[] args) {
        int value = 33;
        System.out.println(add( a: 1, value));
    }
}
```

- Javac 컴파일, javap disassemble 해보기

```
PS C:\Users\jij09\IdeaProjects\COMPILERINTRODUCTION\W10\src> javap -c Test.class
Compiled from "Test.j"
public class Test {
  public Test();
    Code:
       0: aload_0
       1: invokespecial #25                 // Method java/lang/Object."<init>":()V
       4: return

  public static int add(int, int);
    Code:
       0: iload_0
       1: iload_1
       2: iadd
       3: istore_2
       4: iload_2
       5: ireturn

  public static void main(java.lang.String[]);
    Code:
       0: ldc           #30                 // int 33
       2: istore_2
       3: getstatic     #17                 // Field java/lang/System.out:Ljava/io/PrintStream;
       6: ldc           #11                 // int 1
       9: invokestatic  #20                 // Method add:(II)I
      12: invokevirtual #9                  // Method java/io/PrintStream.println:(I)V
      15: return
}
```

```
PS C:\Users\jij09\IdeaProjects\COMPILERINTRODUCTION\W10\src> javap -v Test.class
Classfile /C:/Users/jij09/IdeaProjects/COMPILERINTRODUCTION/W10/src/Test.class
  Last modified 2023. 11. 9.; size 403 bytes
  SHA-256 checksum 61ba5fa939f456833a7018d03b1ef2987a2806f07b53bad66dbed38a1a5c5267
  Compiled from "Test.j"
public class Test
  minor version: 3
  major version: 45
  flags: (0x0021) ACC_PUBLIC, ACC_SUPER
  this_class: #23                         // Test
  super_class: #6                         // java/lang/Object
  interfaces: 0, fields: 0, methods: 3, attributes: 1
Constant pool:
   #1 = NameAndType        #27:#31        // out:Ljava/io/PrintStream;
   #2 = Utf8               ([Ljava/lang/String;)V
   #3 = Utf8               java/lang/Object
   #4 = Utf8               Test
   #5 = Utf8               <init>
   #6 = Class              #3             // java/lang/Object
   #7 = NameAndType        #5:#12         // "<init>":()V
   #8 = Class              #21            // java/io/PrintStream
   #9 = Methodref          #8.#28         // java/io/PrintStream.println:(I)V
  #10 = Utf8               Test.j
  #11 = Integer            1
  #12 = Utf8               ()V
  #13 = Class              #26            // java/lang/System
  #14 = Utf8               Code
  #15 = Utf8               main
  #16 = Utf8               (II)I
  #17 = Fieldref           #13.#1         // java/lang/System.out:Ljava/io/PrintStream;
  #18 = Utf8               SourceFile
  #19 = Utf8               (I)V
  #20 = Methodref          #23.#29        // Test.add:(II)I
  #21 = Utf8               java/io/PrintStream
  #22 = Utf8               println
  #23 = Class              #4             // Test
  #24 = Utf8               add
  #25 = Methodref          #6.#7          // java/lang/Object."<init>":()V
  #26 = Utf8               java/lang/System
  #27 = Utf8               out
       8: invokestatic  #13                  // Method add:(II)I
      11: invokevirtual #19                  // Method java/io/PrintStream.println:(I)V
      14: return
}
```

3. jasmin 으로 .class 파일로 변환 javap disassemble

```
jij09@DESKTOP-ESOMH0M MINGW64 ~/IdeaProjects/COMPILERINTRODUCTION/jasmin-2.4 (ma
in)
$ java -jar jasmin.jar Test.j
Generated: Test.class
```

Test.class

Decompiled .class file, bytecode version: 45.3 (Java 1.1)

```
//
// Source code recreated from a .class file by IntelliJ IDEA
// (powered by FernFlower decompiler)
//

public class Test {
    public Test() {
    }

    public static int add(int var0, int var1) {
        int var2 = var0 + var1;
        return var2;
    }

    public static void main(String[] var0) {
        byte var2 = 33;
        System.out.println(add(1, var2));
    }
}
```

```
PS C:\Users\jij09\IdeaProjects\COMPILERINTRODUCTION\W10\src> javap -c Test.class
Compiled from "Test.j"
public class Test {
  public Test();
    Code:
       0: aload_0
       1: invokespecial #25                 // Method java/lang/Object."<init>":()V
       4: return

  public static int add(int, int);
    Code:
       0: iload_0
       1: iload_1
       2: iadd
       3: istore_2
       4: iload_2
       5: ireturn

  public static void main(java.lang.String[]);
    Code:
       0: ldc           #30                 // int 33
       2: istore_2
       3: getstatic     #17                 // Field java/lang/System.out:Ljava/io/PrintStream;
       6: ldc           #11                 // int 1
       9: invokestatic  #20                 // Method add:(II)I
      12: invokevirtual #9                  // Method java/io/PrintStream.println:(I)V
```

```
PS C:\Users\jij09\IdeaProjects\COMPILERINTRODUCTION\W10\src> javap -v Test.class
Classfile /C:/Users/jij09/IdeaProjects/COMPILERINTRODUCTION/W10/src/Test.class
  Last modified 2023. 11. 9.; size 403 bytes
  SHA-256 checksum 61ba5fa939f456833a7018d03b1ef2987a2806f07b53bad66dbed38a1a5c5267
  Compiled from "Test.j"
public class Test
  minor version: 3
  major version: 45
  flags: (0x0021) ACC_PUBLIC, ACC_SUPER
  this_class: #23                         // Test
  super_class: #6                         // java/lang/Object
  interfaces: 0, fields: 0, methods: 3, attributes: 1
Constant pool:
   #1 = NameAndType        #27:#31        // out:Ljava/io/PrintStream;
   #2 = Utf8               ([Ljava/lang/String;)V
   #3 = Utf8               java/lang/Object
   #4 = Utf8               Test
   #5 = Utf8               <init>
   #6 = Class              #3             // java/lang/Object
   #7 = NameAndType        #5:#12         // "<init>":()V
   #8 = Class              #21            // java/io/PrintStream
   #9 = Methodref          #8.#28         // java/io/PrintStream.println:(I)V
  #10 = Utf8               Test.j
  #11 = Integer            1
  #12 = Utf8               ()V
```

```
  #13 = Class              #26            // java/lang/System
  #14 = Utf8               Code
  #15 = Utf8               main
  #16 = Utf8               (II)I
  #17 = Fieldref           #13.#1         // java/lang/System.out:Ljava/io/PrintStream;
  #18 = Utf8               SourceFile
  #19 = Utf8               (I)V
  #20 = Methodref          #23.#29        // Test.add:(II)I
  #21 = Utf8               java/io/PrintStream
  #22 = Utf8               println
  #23 = Class              #4             // Test
  #24 = Utf8               add
  #25 = Methodref          #6.#7          // java/lang/Object."<init>":()V
  #26 = Utf8               java/lang/System
  #27 = Utf8               out
  #28 = NameAndType        #22:#19        // println:(I)V
  #29 = NameAndType        #24:#16        // add:(II)I
  #30 = Integer            33
  #31 = Utf8               Ljava/io/PrintStream;
{
  public Test();
    descriptor: ()V
    flags: (0x0001) ACC_PUBLIC
    Code:
      stack=1, locals=1, args_size=1
         0: aload_0
         1: invokespecial #25                 // Method java/lang/Object."<init>":()V
         9: invokestatic  #20                 // Method add:(II)I
        12: invokevirtual #9                  // Method java/io/PrintStream.println:(I)V
        15: return
```

IV. 결과 및 설명

3-2에서 Test.j를 직접 JAVA 코드로 변환해본 코드를 컴파일한 결과와, 3-3에서 Test.j 파일을 jasmin을 이용하여 .class 파일로 변환한 결과가 같다.

또한 직접 표현한 JAVA코드를 javap -c 명령어로 disassemble한 결과, 주어진 Test.j 코드 중 중괄호 내부에서 스택의 움직임 혹은 호출을 나타내는 부분인 5-7, 12-17, 22-29 코드가 그대로 드러나는 것을 확인할 수 있다.