程序设计方法学——class文件分析

3190105609 汪辉

执行编译指令：

```
$ javac B.java
```

Java中每一个类都会编译得到一个对应的class文件，所以编译`B.java`文件会得到两个class文件，分别是`B.java`中定义的两个类A和B。

以下指令查看`B.class`内容：

```
$ javap -verbose B.class
Classfile /D:/study/▒▒▒▒▒▒▒/2▒▒▒▒▒▒▒3▒▒▒/B.class
  Last modified 2021▒12▒2▒; size 422 bytes
  MD5 checksum 235813af435e98a8a538a0754e7f189c
  Compiled from "B.java"
public class B
  minor version: 0
  major version: 55
  flags: (0x0021) ACC_PUBLIC, ACC_SUPER
  this_class: #7                          // B
  super_class: #8                         // java/lang/Object
  interfaces: 0, fields: 0, methods: 2, attributes: 1
Constant pool:
   #1 = Methodref          #8.#17         // java/lang/Object."<init>":()V
   #2 = Class              #18            // A
   #3 = Methodref          #2.#17         // A."<init>":()V
   #4 = Fieldref           #19.#20        //
java/lang/System.out:Ljava/io/PrintStream;
   #5 = Methodref          #2.#21         // A.f:(I)I
   #6 = Methodref          #22.#23        // java/io/PrintStream.println:(I)V
   #7 = Class              #24            // B
   #8 = Class              #25            // java/lang/Object
   #9 = Utf8               <init>
  #10 = Utf8               ()V
  #11 = Utf8               Code
  #12 = Utf8               LineNumberTable
  #13 = Utf8               main
  #14 = Utf8               ([Ljava/lang/String;)V
  #15 = Utf8               SourceFile
  #16 = Utf8               B.java
  #17 = NameAndType        #9:#10         // "<init>":()V
  #18 = Utf8               A
  #19 = Class              #26            // java/lang/System
  #20 = NameAndType        #27:#28        // out:Ljava/io/PrintStream;
  #21 = NameAndType        #29:#30        // f:(I)I
  #22 = Class              #31            // java/io/PrintStream
  #23 = NameAndType        #32:#33        // println:(I)V
  #24 = Utf8               B
  #25 = Utf8               java/lang/Object
  #26 = Utf8               java/lang/System
  #27 = Utf8               out
  #28 = Utf8               Ljava/io/PrintStream;
```

```
  #29 = Utf8               f
  #30 = Utf8               (I)I
  #31 = Utf8               java/io/PrintStream
  #32 = Utf8               println
  #33 = Utf8               (I)V
{
  public B();
    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 7: 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: new           #2                  // class A
         3: dup
         4: invokespecial #3                  // Method A."<init>":()V
         7: astore_1
         8: getstatic     #4                  // Field
java/lang/System.out:Ljava/io/PrintStream;
        11: aload_1
        12: bipush        10
        14: invokevirtual #5                  // Method A.f:(I)I
        17: invokevirtual #6                  // Method
java/io/PrintStream.println:(I)V
        20: return
      LineNumberTable:
        line 9: 0
        line 10: 8
        line 11: 20
}
SourceFile: "B.java"
```

大致可以分为四个部分：文件头、类信息、class常量池和可执行代码。

- 文件头部分：

  ```
  Classfile /D:/study/▒▒▒▒▒▒/2▒▒▒▒▒▒3▒▒▒/B.class
    Last modified 2021▒▒12▒▒2▒▒; size 422 bytes
    MD5 checksum 235813af435e98a8a538a0754e7f189c
    Compiled from "B.java"
  ```

  记录了文件路径、更改日期、大小、MD5校验和源文件信息。

- 类信息：

```
public class B
  minor version: 0
  major version: 55
  flags: (0x0021) ACC_PUBLIC, ACC_SUPER
  this_class: #7                          // B
  super_class: #8                         // java/lang/Object
  interfaces: 0, fields: 0, methods: 2, attributes: 1
```

minor version、major version分别是此class文件的次版本号和主版本号。

flags字段记录了这个class的访问权限标志，ACC_PUBLIC的值为0x0001，ACC_SUPER的值为0x0020，相与结果即flags(0x0021)。意味着类B可以被外部访问和继承。

this_class指向常量池中第7个常量，从常量池中看到又指向第24个，最终指向字面量类`B`。

super_class指向第8个常量，从第8个指向第25个常量，最终指向的是字面量`java/lang/Object`这个基类。

interfaces字段、fields字段均为0，表示类B既没有实现任何接口，也没有自己的域或者说成员。

methods字段为2，应该是默认构造方法和自定义的main方法。

attributes字段为1，猜测是和类B有关的默认的属性。

- class常量池：

```
Constant pool:
   #1 = Methodref          #8.#17         // java/lang/Object."<init>":()V
   #2 = Class              #18            // A
   #3 = Methodref          #2.#17         // A."<init>":()V
   #4 = Fieldref           #19.#20        //
java/lang/System.out:Ljava/io/PrintStream;
   #5 = Methodref          #2.#21         // A.f:(I)I
   #6 = Methodref          #22.#23        // java/io/PrintStream.println:
(I)V
   #7 = Class              #24            // B
   #8 = Class              #25            // java/lang/Object
   #9 = Utf8               <init>
  #10 = Utf8               ()V
  #11 = Utf8               Code
  #12 = Utf8               LineNumberTable
  #13 = Utf8               main
  #14 = Utf8               ([Ljava/lang/String;)V
  #15 = Utf8               SourceFile
  #16 = Utf8               B.java
  #17 = NameAndType        #9:#10         // "<init>":()V
  #18 = Utf8               A
  #19 = Class              #26            // java/lang/System
  #20 = NameAndType        #27:#28        // out:Ljava/io/PrintStream;
  #21 = NameAndType        #29:#30        // f:(I)I
  #22 = Class              #31            // java/io/PrintStream
  #23 = NameAndType        #32:#33        // println:(I)V
  #24 = Utf8               B
  #25 = Utf8               java/lang/Object
  #26 = Utf8               java/lang/System
  #27 = Utf8               out
  #28 = Utf8               Ljava/io/PrintStream;
  #29 = Utf8               f
  #30 = Utf8               (I)I
  #31 = Utf8               java/io/PrintStream
```

```
  #32 = Utf8               println
  #33 = Utf8               (I)V
```

常量池中：

- `#9 = Utf8`表明常量9是一个字面量常量，内容就是后面的`<init>`。在编译时方法名、整型字符串常量以及final修饰的变量的字面量值等都会记录在这样的常量中。
- `#5 = Methodref`表明常量5是一个方法的名字，该方法名字的实际内容需要通过指向另一个类及其方法的字面量来找到。比如这里常量5指向常量2——类B，的常量21——方法`f:(I)I`。
- `#4 = Fieldref`表明常量4是一个域的名字，其引用找到其实际内容的方法和`Methodref`一样。
- `#2 = Class`表明常量2是一个类，它的名字通过引用另一个字面量常量得到。
- `#17 = NameAndType`表明常量17是一个带参数列表名的方法名。