Class文件分析

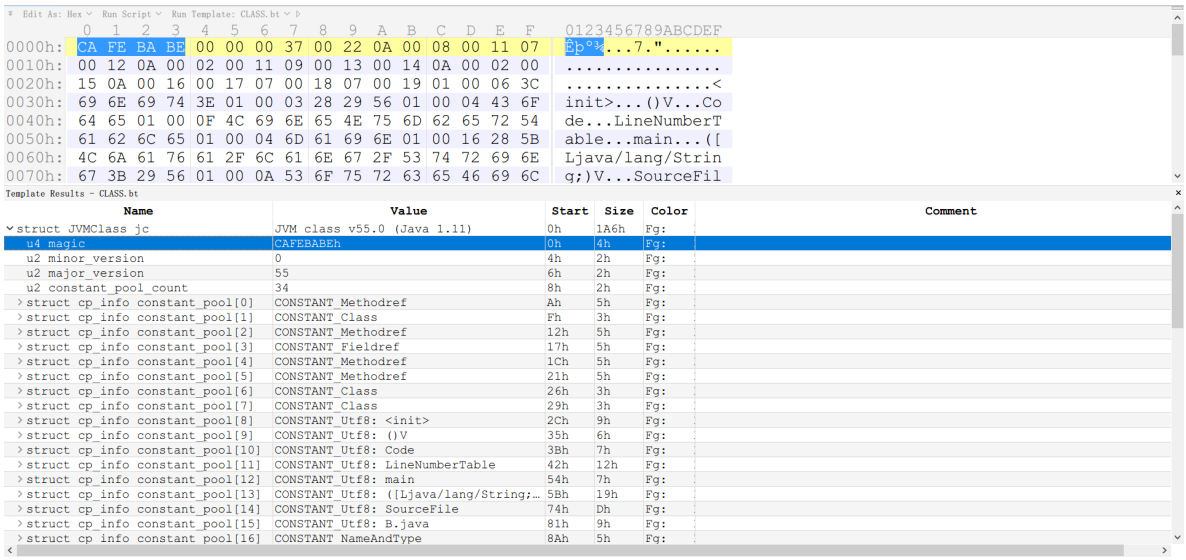
B.java源代码

```
class A {
    int f(int k) {
        return k*2;
    }
}

public class B {
    public static void main(String[] args) {
        A a = new A();
        System.out.println(a.f(10));
    }
}
```

javac编译后产生A.class和B.class两个文件

010editor打开B.class



```
Edit As: Hex  Run Script  Run Template: CLASS.bt

          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F    0123456789ABCDEF
0000h:   CA FE BA BE 00 00 00 37 00 22 0A 00 08 00 11 07    Êþº¾...7."......
0010h:   00 12 0A 00 02 00 11 09 00 13 00 14 0A 00 02 00    ................
0020h:   15 0A 00 16 00 17 07 00 18 07 00 19 01 00 06 3C    ...............<
0030h:   69 6E 69 74 3E 01 00 03 28 29 56 01 00 04 43 6F    init>...()V...Co
0040h:   64 65 01 00 0F 4C 69 6E 65 4E 75 6D 62 65 72 54    de...LineNumberT
0050h:   61 62 6C 65 01 00 04 6D 61 69 6E 01 00 16 28 5B    able...main...([
0060h:   4C 6A 61 76 61 2F 6C 61 6E 67 2F 53 74 72 69 6E    Ljava/lang/Strin
0070h:   67 3B 29 56 01 00 0A 53 6F 75 72 63 65 46 69 6C    g;)V...SourceFil

Template Results - CLASS.bt

Name                                    Value                                   Start   Size    Color   Comment
struct JVMClass jc                      JVM class v55.0 (Java 1.11)             0h      1A6h    Fg:
  u4 magic                              CAFEBABEh                               0h      4h      Fg:
  u2 minor_version                      0                                       4h      2h      Fg:
  u2 major_version                      55                                      6h      2h      Fg:
  u2 constant_pool_count                34                                      8h      2h      Fg:
  struct cp_info constant_pool[0]       CONSTANT_Methodref                      Ah      5h      Fg:
  struct cp_info constant_pool[1]       CONSTANT_Class                          Fh      3h      Fg:
  struct cp_info constant_pool[2]       CONSTANT_Methodref                      12h     5h      Fg:
  struct cp_info constant_pool[3]       CONSTANT_Fieldref                       17h     5h      Fg:
  struct cp_info constant_pool[4]       CONSTANT_Methodref                      1Ch     5h      Fg:
  struct cp_info constant_pool[5]       CONSTANT_Methodref                      21h     5h      Fg:
  struct cp_info constant_pool[6]       CONSTANT_Class                          26h     3h      Fg:
  struct cp_info constant_pool[7]       CONSTANT_Class                          29h     3h      Fg:
  struct cp_info constant_pool[8]       CONSTANT_Utf8: <init>                   2Ch     9h      Fg:
  struct cp_info constant_pool[9]       CONSTANT_Utf8: ()V                      35h     6h      Fg:
  struct cp_info constant_pool[10]      CONSTANT_Utf8: Code                     3Bh     7h      Fg:
  struct cp_info constant_pool[11]      CONSTANT_Utf8: LineNumberTable          42h     12h     Fg:
  struct cp_info constant_pool[12]      CONSTANT_Utf8: main                     54h     7h      Fg:
  struct cp_info constant_pool[13]      CONSTANT_Utf8: ([Ljava/lang/String;...  5Bh     19h     Fg:
  struct cp_info constant_pool[14]      CONSTANT_Utf8: SourceFile               74h     Dh      Fg:
  struct cp_info constant_pool[15]      CONSTANT_Utf8: B.java                   81h     9h      Fg:
  struct cp_info constant_pool[16]      CONSTANT_NameAndType                    8Ah     5h      Fg:
```

头四个字节CAFEBABE是class 的头文件表示

minor_version为0, major_version为55, 表示JVM版本为11(由jdk11编译出)

```
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
```

常量池

常量有34个(实际上是33个), javap查看也可以看到一共33个常量

```
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
```

第1个常量是一个Methodref，表示一个没有参数，返回值为void的构造函数的调用，属于Object，是类在没有自定义默认构造函数的情况下调用JVM为其分配的父类Object的构造函数

```
1  #1 = Methodref          #8.#17         // java/lang/Object."<init>":()V
```

第2个常量是一个Class，表示类A

```
1  #2 = Class              #18            // A
```

第3个常量是一个Methodref，表示类A的构造函数的调用

```
1  #3 = Methodref          #2.#17         // A."<init>":()V
```

第4个常量是一个Fieldref，表示printStream

```
1  #4 = Fieldref           #19.#20        //
   java/lang/System.out:Ljava/io/PrintStream;
```

第5个常量是一个Methodref，表示类A的方法f的调用，接收一个int参数，返回一个int

```
1  #5 = Methodref          #2.#21         // A.f:(I)I
```

第6个常量是一个Methodref，表示System.out.println方法调用

```
1  #6 = Methodref          #22.#23        // java/io/PrintStream.println:(I)V
```

第7个常量是一个Class，表示类B

```
1  #7 = Class              #24            // B
```

第8个常量是一个Class，表示类Object

```
1  #8 = Class              #25            // java/lang/Object
```

第9-16个常量都是Utf8，是字面量

```
 #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
```

第17个常量是一个NameAndType，表示一个构造函数是一个没有参数且返回值是void的函数

```
#17 = NameAndType        #9:#10         // "<init>":()V
```

第18个常量是一个Utf8，字面量

```
#18 = Utf8               A
```

第19个常量是一个Class，表示类System

```
#19 = Class              #26            // java/lang/System
```

第20个常量是一个NameAndType，表示out是一个PrintStream

```
#20 = NameAndType        #27:#28        // out:Ljava/io/PrintStream;
```

第21个常量是一个NameAndType，表示f是一个接受一个int参数返回一个int的函数

```
#21 = NameAndType        #29:#30        // f:(I)I
```

第22个常量是一个Class，表示类PrintStream

```
#22 = Class              #31            // java/io/PrintStream
```

第23个常量是一个NameAndType，表示println是接收一个int参数返回void的函数

```
#23 = NameAndType        #32:#33        // println:(I)V
```

第24-33个常量都是Utf8，是字面量

```
#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
```

常量池后是access_info, 以及method_info

Name	Value	Start	Size	Color	Comment
struct cp_info constant_pool[27]	CONSTANT_Utf8: Ljava/…	D8h	18h	Fg:	
struct cp_info constant_pool[28]	CONSTANT_Utf8: f	F0h	4h	Fg:	
struct cp_info constant_pool[29]	CONSTANT_Utf8: (I)I	F4h	7h	Fg:	
struct cp_info constant_pool[30]	CONSTANT_Utf8: java/i…	FBh	16h	Fg:	
struct cp_info constant_pool[31]	CONSTANT_Utf8: println	111h	Ah	Fg:	
struct cp_info constant_pool[32]	CONSTANT_Utf8: (I)V	11Bh	7h	Fg:	
enum AF access_flags	33: ACC_PUBLIC, ACC_P…	122h	2h	Fg:	
u2 this_class	7	124h	2h	Fg:	
u2 super_class	8	126h	2h	Fg:	
u2 interfaces_count	0	128h	2h	Fg:	
u2 fields_count	0	12Ah	2h	Fg:	
u2 methods_count	2	12Ch	2h	Fg:	
struct method_info methods[0]		12Eh	2Bh	Fg:	
enum AF access_flags	1: ACC_PUBLIC, ACC_PR…	12Eh	2h	Fg:	
u2 name_index	9	130h	2h	Fg:	
u2 descriptor_index	10	132h	2h	Fg:	
u2 attributes_count	1	134h	2h	Fg:	
struct attribute_info attributes		136h	23h	Fg:	
struct method_info methods[1]		159h	43h	Fg:	
enum AF access_flags	9: ACC_PUBLIC, ACC_PR…	159h	2h	Fg:	
u2 name_index	13	15Bh	2h	Fg:	
u2 descriptor_index	14	15Dh	2h	Fg:	
u2 attributes_count	1	15Fh	2h	Fg:	

```
{
  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"
```