**Jvm**

# Class文件结构

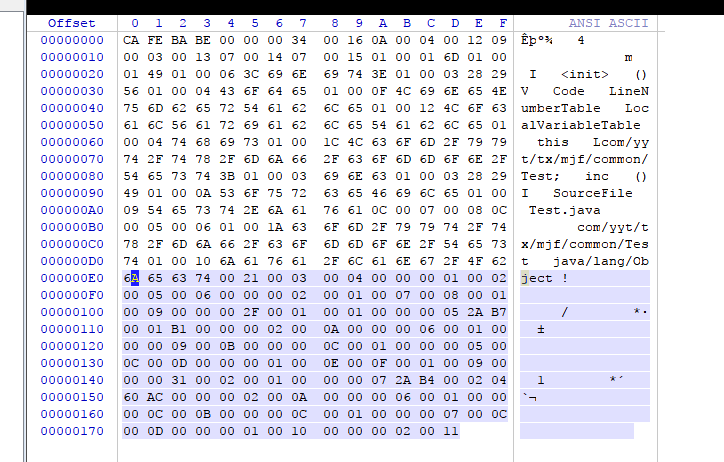
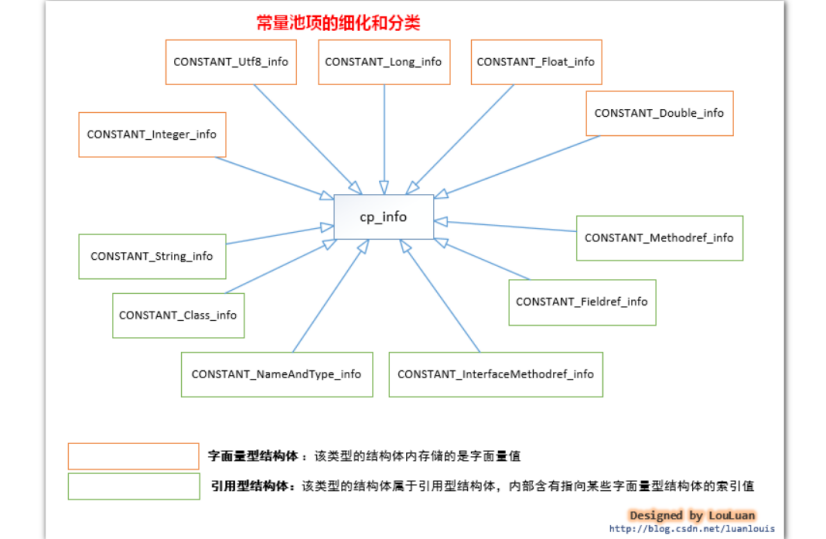




图2



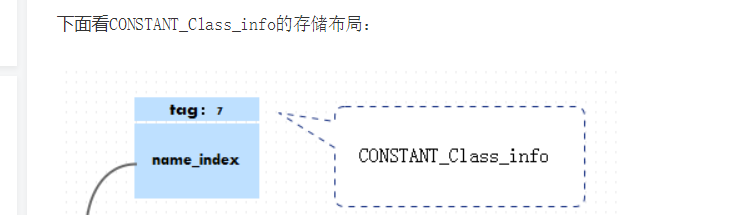
图三



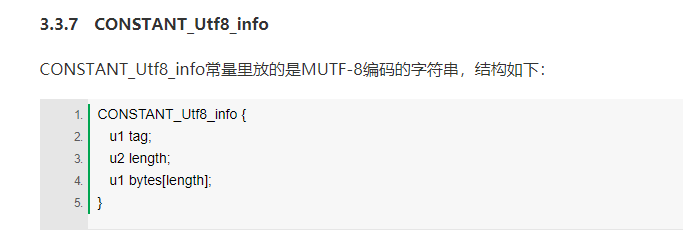
图五



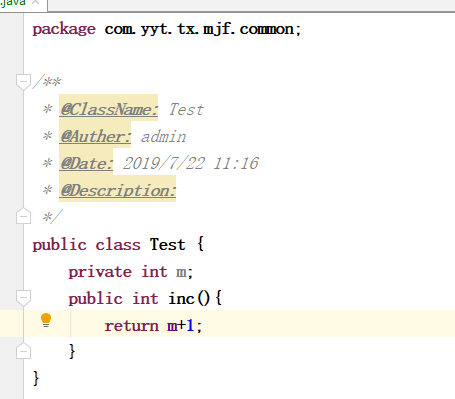
图六



图七



图八



图九



图十



图11

## 常量池

1. 前四个字节0xCAFEBABE代表魔法数，它的作用是确定这个文件是否为一个能被虚拟机接受的class文件，起到省份识别的作用.
2. 第五字节和第六字节表示副版本号0x0000，第七字节和第八字节表示主版本号0x0034,就是十进制的52
3. 第九和第十个字节表示常量池的计数器，即0x0016，十进制为22，也就是常量池中有21个常量，索引值范围1-21，注意:虽然值为 0 的 constant\_pool 索引是无效的，但其他用到常量池的数据结构可以使用索引 0 来表示“不引用任何一个常量池项”的意思。
4. 常量池中的第一项，偏移地址(0x0000000A) 0x0A，即十进制10，查看图2，代表类中的方法(CONSTANT\_methodref\_info),0x0004代表class\_index，0x0012(十进制18)，表示name\_and\_type\_Index,指向第十八项常量



1. 常量池中的第二项,tag=0x00，代表【CONSTANT\_FieldRef\_INFO】【字段引用】,偏移量(0x00000010)0x0003，指向常量池中的第三项常量,偏移量0x00000012【0x0013】(十进制为19，指向常量池第十九项常量)
2. 常量池中的第三项,偏移量0x00000014【0x07】,表示CONSTANT\_class\_info(类或者接口的全限定名称),后面两个字节0x0014(十进制为20,指向向量池第20个常量)
3. 常量池中的第四项,偏移量0x00000017【0x07】，表示类或者接口的权限定名称,name\_index为0x0015(十进制为21)
4. 第五项，偏移量0x0000001A【0x01】,表示字符串常量。后面两个字节代表字符串的长度(0x0001，长度为1),后面一个字节代表bytes【0x6D,十进制为6\*16+13=109】,代表小谢字母m



1. 第六项:偏移量0x0000001E【0x01】,表示字符串,长度为【0x0001】，bytes为【0x49,十进制为4\*16+9=73】,表示字母I



1. 第七项:偏移量0x00000022【0x01】，表示字符串，长度为【0x0006】，bytes【0x3C696E69743E】，即字符串<init>,具体查看ASCII(<https://baike.baidu.com/item/ASCII/309296?fr=aladdin>)
2. 第八项:偏移量0x0000002B【0x01】表示字符串常量，长度为0x0003,bytes【0x282956】,即字符串()V
3. 第九项:偏移量0x00000031【0x01】表示字符串常量,长度【0x0004】，bytes【0x436F6465】,即字符串【Code】
4. 第十项:偏移量0x00000038【0x01】表示字符串常量,长度【0x000F】,bytes【0x4C696E654E756D6265725461626C65】即字符串【LineNumberTable】
5. 第十一项:偏移量0x0000004A【0x01】表示字符串常量,长度【0x0012】，bytes【0x4C6F63.......】，代表LocalVariableTable
6. 第十二项：偏移量0x0000005F【0x01】表示字符串常量，长度【0x0004】,bytes【0x74686973】，表示字符串this
7. 第十三项：0x0000066【0x01】，长度【0x001C】,十进制为28，bytes【...........】
8. 第十四项:0x00000085【0x01】，长度【0x0003】，bytes【0x696E63】inc
9. 第十五项:0x0000008B【0x01】，长度【0x0003】，bytes【0x282949】()I
10. 第十六项:0x00000091【0x01】，长度【0x000A】，bytes【】【SourceFile】
11. 第十七项:0x0000009E【0x01】，长度【0x0009】，bytes【】【Test.java】
12. 十八项:0x000000AA【0x0C】，代表CONSTANT\_ NameAndType\_Info,name\_index【0x0007】,descriptor\_index【0x0008】
13. 十九项:0x000000AF【0x0C】，代表CONSTANT\_ NameAndType\_Info，name\_index【0x0005】，descriptor\_index【0x0006】
14. 二十项:0x000000B4【0x01】，长度【0x001A】【十进制26】,即com/yyt/tx/mjf/common/test
15. 二十一项:0x000000D1【0x01】,长度【0x10】，即java/lang/Object

## 访问标志

偏移量0x000000E4【0x0021】=0x0001|0x0020表示该类是public的。

## 类索引

偏移量0x000000E6【0x0003】，指向常量池中第三个常量

## 父类的索引

偏移量0x000000E8【0x0004】，指向常量池中第四个常量，java/lang/Object

## 接口计数器

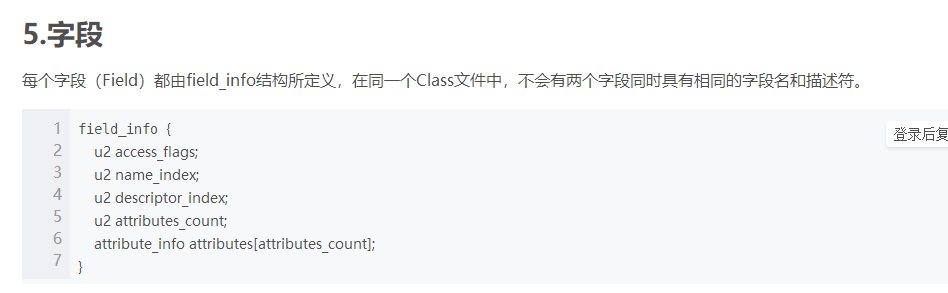
偏移量0x000000EA【0x000000】，数量为0

## 接口表

## 字段计数器

偏移量0x000000EC【0x0001】，数量为1

## 字段表





访问标志：偏移量0x000000EE【0x0002】代表private

Name\_index:偏移量0x000000F0【0x0005】指向常量池的第五项常量m

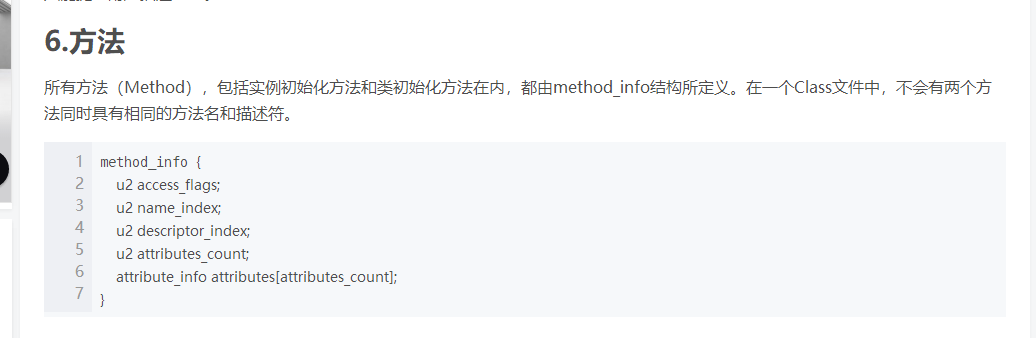
Descriptor\_index：偏移量0x000000F2【0x0006】指向常量池中的第六项常量I

Attribute\_count：偏移量【0x000000F4】【0x0000】属性数量为0

## methods\_count

偏移量【0x000000F6】【0x0002】，有两个方法

## 方法表





1. 偏移量:0x000000F8
2. Access\_flags【0x0001】表示该方法是公有的
3. Name\_index：偏移量【0x000000FA】【0x0007】，代表<init>
4. Descriptor\_index：偏移量【0x000000FC】【0x00008】，代表()V，即第八项常量
5. Attributes\_count:偏移量【0x000000FE】【0x0001】

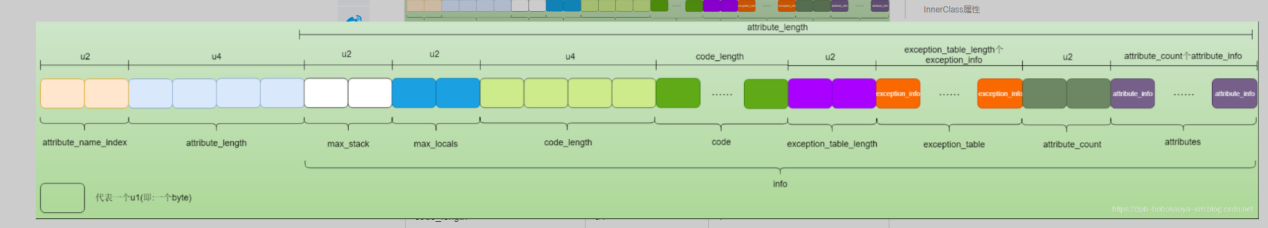
### 属性表集合

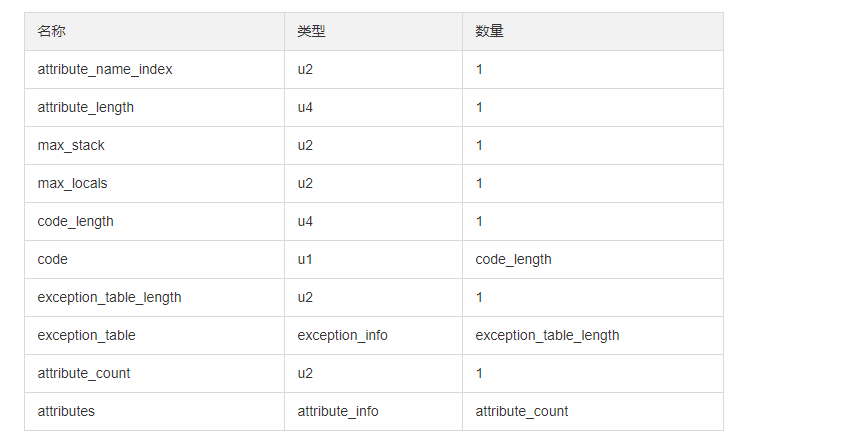


偏移量:【0x00000100】【0x0009】,指向常量池第九个常量:Code.

Attribute\_length:偏移量【0x0000002F】

#### code属性表的结构





Max\_stack:【0x0001】，最大栈为1.

Max\_locals:偏移量为0x00000108【0x0001】，本地变量为1个

Code\_length:偏移量为0x0000010A【0x00000005】

Code：偏移量0x0000010E【0x2AB70001B1】

字节码指令表:<https://www.jianshu.com/p/7d623f441bed>

0x2A表示

0xB7







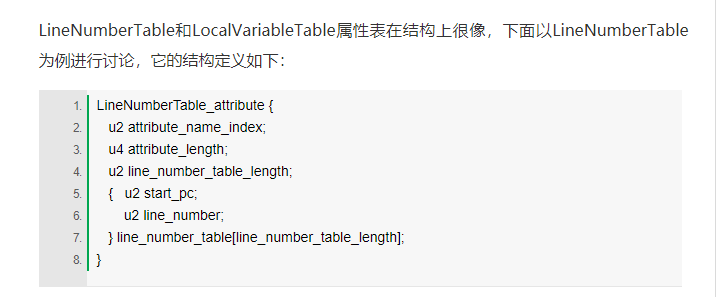


**Exception\_table\_length:偏移量0x00000113【0x0000，表示长度为0】**

**Attribute\_count:偏移量0x00000115【0x0002，表示两个属性】**

**第一个属性：**

**Attribute\_name\_index:偏移量0x00000117【0x000A，指向第常量池中第十个常量LineNumberTable】**



**Attribute\_length:偏移量0x00000119【0x0000006】**

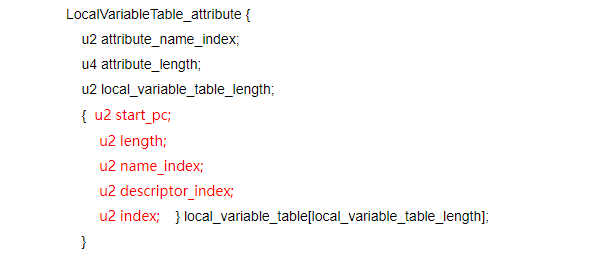
**Line\_number\_table\_length:偏移量0x0000011D【0x0001】**

**Start\_pc(字节码的行号):偏移量0x0000011F【0x0000,行号为0】**

**Line\_number(源码的行号):偏移量0x00000121【0x0009，第九行】**

**接下来就是第二个属性了**

1. **Attribute\_name\_index:偏移量0x00000123【0x000B，指向常量池中的第十一个属性（LocalVariableTable）】**
2. **Attribute\_length:偏移量0x00000125【0x00000C】**



1. **Local\_variable\_table\_length**:偏移量0x00000129【0x0001，只有一个本地变量】
2. Start\_pc:偏移量0x0000012B【0x0000】
3. Length:偏移量0x0000012D【0x0005】
4. Name\_index:偏移量0x0000012F【0x000C，指向第十二个常量(this)】
5. Descriptior\_index:0x00000131【0x000D，指向第十三个常量】
6. Index:偏移量0x00000133【0x0000】

## 第二个方法

### 结构

1. 偏移量:0x00000135
2. Access\_flags【0x0001】表示该方法是公有的
3. Name\_index：偏移量【0x00000137】【0x000E】，代表方法inc
4. Descriptor\_index：偏移量【0x00000139】【0x0000F】，代表()I，即第八项常量
5. Attributes\_count:偏移量【0x00000013B】【0x0001】,有一个属性

### 属性

偏移量:0x0000013D【0x0009】,指向常量池第九个常量:Code.

Attribute\_length:偏移量【0x00000031】

Max\_stack:【0x0002】，最大栈为2.

Max\_locals:偏移量为0x00000145【0x0001】，本地变量为1个

Code\_length:偏移量为0x00000147【0x00000007】

Code：偏移量0x0000014B【0x2AB400020460AC】，去指令表中查即可







**Exception\_table\_length:偏移量0x00000152【0x0000，表示长度为0】**

**Attribute\_count:偏移量0x00000154【0x0002，表示两个属性】**

**第一个属性：**

**Attribute\_name\_index:偏移量0x00000156【0x000A，指向第常量池中第十个常量LineNumberTable】**

**Attribute\_length:偏移量0x00000158【0x0000006】**

**Line\_number\_table\_length:偏移量0x000001【0x0001】**

**Start\_pc(字节码的行号):偏移量0x0000015E【0x0000,行号为0】**

**Line\_number(源码的行号):偏移量0x00000160【0x000C，第十二行】**

**接下来就是第二个属性了**

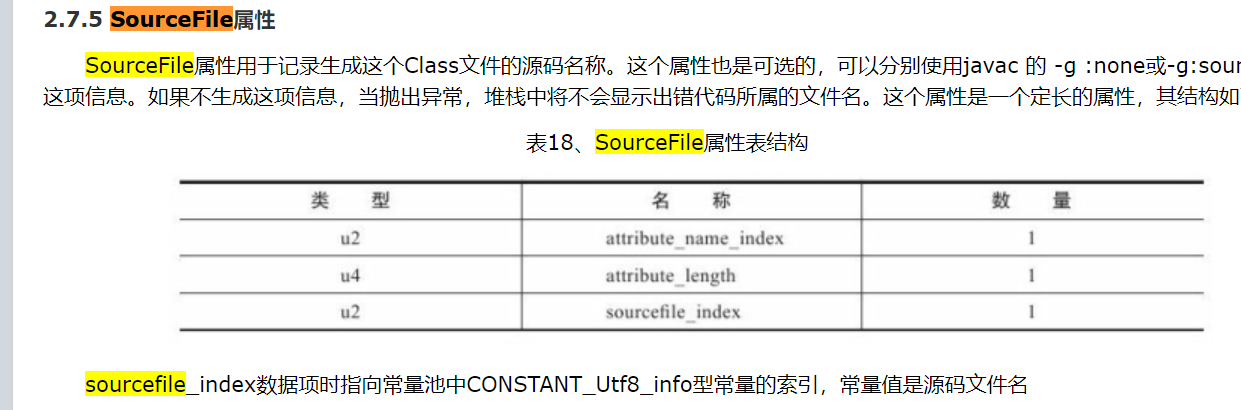
1. **Attribute\_name\_index:偏移量0x00000162【0x000B，指向常量池中的第十一个属性（LocalVariableTable）】**
2. **Attribute\_length:偏移量0x00000164【0x00000C】**
3. **Local\_variable\_table\_length**:偏移量0x00000168【0x0001，只有一个本地变
4. Start\_pc:偏移量0x0000016B【0x0000】
5. Length:偏移量0x0000016C【0x0007】
6. Name\_index:偏移量0x0000016E【0x000C，指向第十二个常量(this)
7. Descriptior\_index:0x00000170【0x000D，指向第十三个常量】
8. Index:偏移量0x00000172【0x0000】

方法到此结束！！！！！！

接下来就是属性

属性数量:偏移量0x00000174【0x0001】

属性名称索引:0x0010，指向第16个常量SourceFile



Length:0x00000002

Sourcefile\_index：偏移量0x0000017D【0x0011，指向第17个常量(Test.java)】

**所有的分析到此结束！！！！！！！**