

## Class 文件结构

- 魔数 u4
- 版本 2个u2 次版本号 主版本号
- 常量池个数 u2 是从1开始, 比如22, 就有21个常量
- 常量池信息
- 类访问标志 u2
- 类索引 u2
- 超类索引 u2
- 接口表索引 2+n
- 字段表
- 方法表
- 属性表

## 常量池

CONSTANT_Utf8_info	1
CONSTANT_Integer_info	3
CONSTANT_Float_info	4
CONSTANT_Long_info	5
CONSTANT_Double_info	6
CONSTANT_Class_info	7
CONSTANT_String_info	8
CONSTANT_Fieldref_info	9
CONSTANT_Methodref_info	10
CONSTANT_InterfaceMethodref_info	11
CONSTANT_NameAndType_info	12
CONSTANT_MethodHandle_info	15
CONSTANT_MethodType_info	16
CONSTANT_InvokeDynamic_info	18

每种常量的结构, 开头都是u1的tag位, 表示是那种常量。

```
CONSTANT_Integer_info {  
    u1 tag;  
    u4 bytes;  
}
```

```
CONSTANT_Class_info {  
    u1 tag;
```

```
    u2 name_index;
}
```

## 访问标志

```
ACC_PUBLIC 0001
ACC_FINAL  0010
ACC_SUPER  0020
ACC_INTERFACE 0200
ACC_ABSTRACT 0400
ACC_SYNTHETIC 1000
ACC_ANNOTATION 2000
ACC_ENUM 4000
```

## 字段表

```
{
    u2 fields_count;
    field_info fields[fields_count];
}

field_info{
    u2 access_flag
    u2 name_index
    u2 descriptor_index 类型信息
    u2 attribute_count
    attribute_info attribute_info[attribute_count]
}
```

## 方法表

```
{
    u2 method_count
    method_info method[method_count]
}

method_info {
    u2 access_flags
    u2 name_index 方法名
    u2 descriptor_index 方法属性 int String void....
    u2 attributes_count 属性数量 一般没有
    attribute_info Code_attribute //不占字节，开始的字节属于Code_attribute
}
```

## Code

```
Code_attribute {
    u2 attribute_name_index
    u4 attribute_length //不包含attribute_name_index 和 attribute_length
    u2 max_stack
    u2 max_locals
    u4 code_length
    u1 code_info[code_length]
    u2 exception_table_length{
        u2 start_pc
        u2 end_pc
        u2 handler_pc
        u2 catch_type
    }
    u2 attribute_count
    attribute_info[attribute_count]LineNumberTable、LocalVariableTable
}
```

## LineNumberTable

行号与字节码的对应关系

```
{
    u2 attribute_name_index
    u4 attribute_length 长度所指的不包含attribute_name_index和attribute_length
    {
        u2 有几对映射
        u2 start_pc 字节码的中的行号
        u2 line_number 源码中的行号
    }
}
```