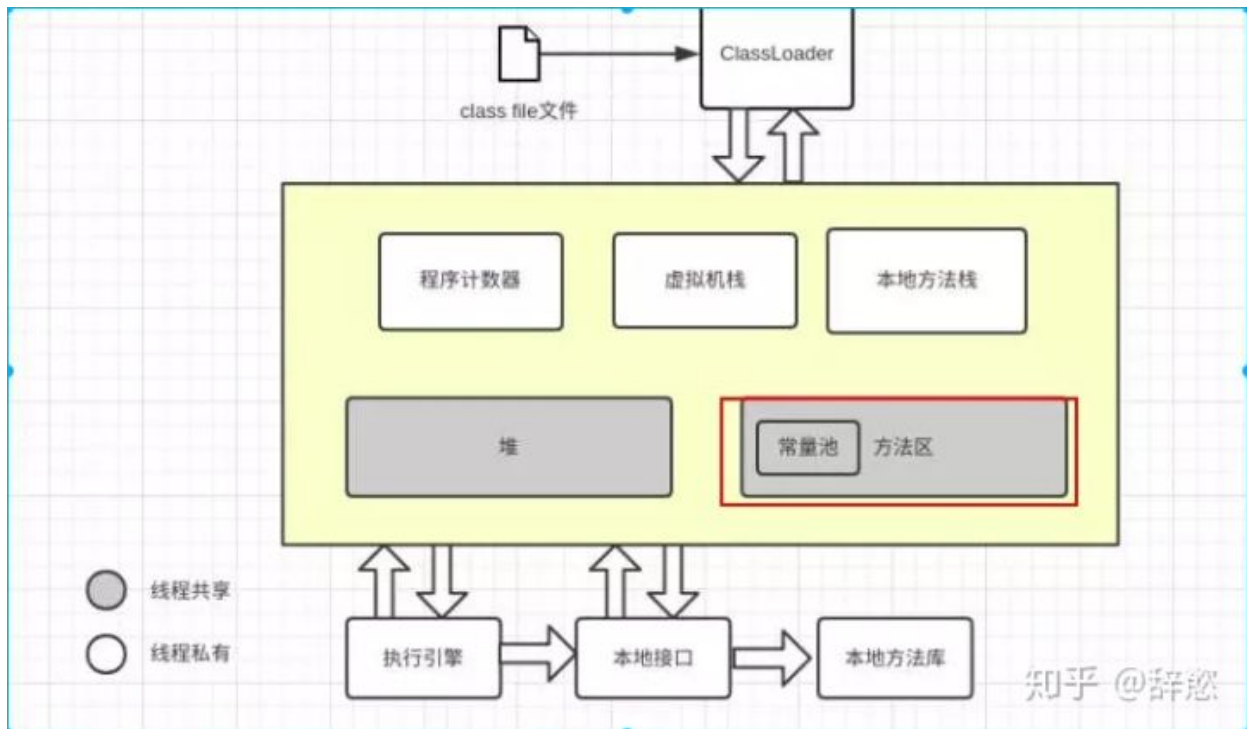


首先，来两个问题

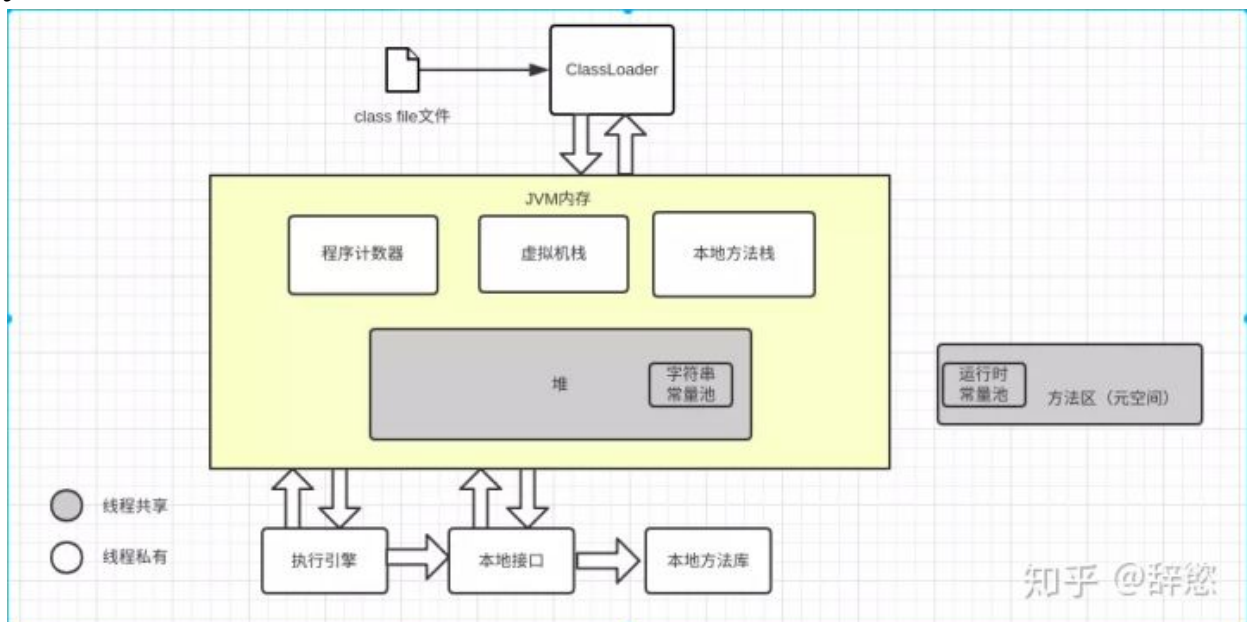
- 1.String为什么是final修饰的?
- 2.String s="ss";和String s=new String("ss");有什么不同;

String类是我们使用频率非常高的一种对象类型。JVM为了提升性能和减少内存开销，避免字符串的重复创建，其维护了一块特殊的内存空间，这就是我们今天要讨论的核心，即字符串池（String Pool）。

java7虚拟机内存模型



java8虚拟机内存模型



创建对象的两种方式：

- 1) 采用字面值的方式赋值
- 2) 采用new关键字新建一个字符串对象。

接下来写一段代码，这段代码大家都知道结果，但是我们今天会讨论背后的原理！

```
1 public static void main(String[] args)
2 {
3     String s1="aa";
4     String s2="aa";
5     String s3=new String("aa");
6     String s4=new String ("aa");
7
8     System.out.println("s1==s2:"+ (s1==s2));
9     System.out.println("s3==s4:"+ (s3==s4));
10 }
11 -----
12 s1==s2:true
13 s3==s4:false
```

### 采用字面值的方式赋值

采用字面值的方式创建一个字符串时，JVM首先会去字符串池中查找是否存在"aa"，如果不存在，则在字符串池中创建"aa"，然后将池中"aa"的引用地址返回给字符串常量s1，这样s1会指向池中"aa"这个字符串对象；如果存在，则不创建任何对象，直接将池中"aa"这个对象的地址返回，赋给s2字符串常量。

这时候解释为什么String是final，当创建"aa"后，aa属于s1，s2共享的内容，当s2要用"aa"时，s1修改了"aa"为"ab"，这程序员得气得半死。

### 采用new关键字新建对象

采用new关键字新建对象时，JVM首先在字符串池中查找有没有"aa"这个字符串对象，如果有，则不在池中再去创建"aa"这个对象了，直接在堆中创建一个"aa"字符串对象，然后将堆中的这个"aa"对象的地址返回赋给引用s3，这样，s3就指向了堆中创建的这个"aa"字符串对象；如果没有，则首先在字符串池中创建一个"aa"字符串对象，然后再在堆中创建一个"aa"字符串对象，然后将堆中这个"aa"字符串对象的地址返回赋给s3引用，这样，s3指向了堆中创建的这个"aaa"字符串对象。

### intern方法

这里有个方法是堆上String对象与字符串池之间的桥梁intern ()

```
1 /**
2  * Returns a canonical representation for the string object.
3  * <p>
4  * A pool of strings, initially empty, is maintained privately by the
5  * class {@code String}.
6  * <p>
7  * When the intern method is invoked, if the pool already contains a
8  * string equal to this {@code String} object as determined by
```

```

9  * the {@link #equals(Object)} method, then the string from the pool is
10 * returned. Otherwise, this {@code String} object is added to the
11 * pool and a reference to this {@code String} object is returned.
12 * <p>
13 * It follows that for any two strings {@code s} and {@code t},
14 * {@code s.intern() == t.intern()} is {@code true}
15 * if and only if {@code s.equals(t)} is {@code true}.
16 * <p>
17 * All literal strings and string-valued constant expressions are
18 * interned. String literals are defined in section 3.10.5 of the
19 * <cite>The Java&trade; Language Specification</cite>.
20 *
21 * @return a string that has the same contents as this string, but is
22 * guaranteed to be from a pool of unique strings.
23 */
24 public native String intern();

```

翻译一下注释的意思，就是在堆上创建一个新对象调用intern () 方法，如果字符串池里没有equals为真的值，就放入字符串池中，把字符串池的引用返回；如果已经存在了直接返回字符串池中的引用。

## 安排代码

```

1  public static void main(String[] args)
2  {
3      String s1="aa";
4      String s2="aa";
5      String s3=new String("aa");
6      String s4=new String ("aa");
7
8      System.out.println("s1==s2:"+s1==s2);
9      System.out.println("s2==s3:"+s2==s3);
10     System.out.println("s2==s3.intern():"+(s2==s3.intern()));
11     System.out.println("s3==s4:"+s3==s4);
12 }
13 -----
14 s1==s2:true
15 s2==s3:false
16 s2==s3.intern():true
17 s3==s4:false

```

