简单说intern方法就是判断并将字符串是否存在于字符串常量池，如果不存在则创建，存在则返回。

1. 分类

静态常量池和动态常量池的关系以及区别

静态常量池

存储的是当class文件被java虚拟机加载进来时存放在方法区的一些字面量和符号引用，字面量包括字符串，基本类型的常量，符号引用其实引用的就是常量池里面的字符串，但符号引用不是直接存储字符串，而是存储字符串在常量池里的索引。

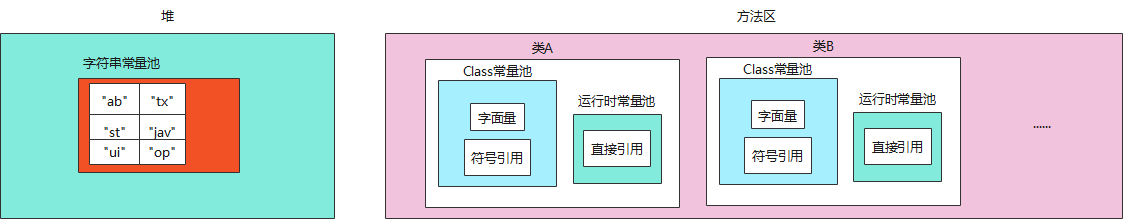
动态常量池

当class文件被加载完成后，java虚拟机会将静态常量池里的内容转移到动态常量池里，在静态常量池的符号引用有一部分是会被转变为直接引用的。

比如说类的静态方法或私有方法，实例构造方法，父类方法，这是因为这些方法不能被重写其他版本，所以能在加载的时候就可以将符号引用转变为直接引用。

而其他的一些方法是在这个方法被第一次调用的时候才会将符号引用转变为直接引用的。

1. JVM常量池



1. 字符串常量池
2. 含义：

在HotSpot VM里实现的string pool功能的是一个StringTable类，它是一个Hash表，默认值大小长度是1009；这个StringTable在每个HotSpot VM的实例只有一份，被所有的类共享。字符串常量由一个一个字符组成，放在了StringTable上。

1. 内容：

JDK6.0及之前版本中：String Pool里放的都是字符串常量。

JDK6.0后：由于String#intern()发生了改变，因此String Pool中也可以存放放于堆内的字符串对象的引用（字符串常量+字符串对象引用）。

1. 位置：

在JDK6.0及之前版本，字符串常量池是放在Perm Gen区(也就是方法区)中；

在JDK7.0版本，字符串常量池被移到了堆中了。至于为什么移到堆内，大概是由于方法区的内存空间太小了。

1. 数量：

只有一个，所有类共享

1. Class常量池
2. 含义：

我们写的每一个Java类被编译后，就会形成一份class文件；class文件中除了包含类的版本、字段、方法、接口等描述信息外，还有一项信息就是常量池，用于存放编译器生成的各种字面量(Literal)和符号引用(Symbolic References)；

每个class文件都有一个class常量池。

1. 内容：

字面量、符号引用

字面量包括：1.文本字符串 2.八种基本类型的值 3.被声明为final的常量等。

符号引用包括：1.类和方法的全限定名 2.字段的名称和描述符 3.方法的名称和描述符。

1. 位置：

方法区。

1. 数量：

每个class一个，即每个类一个。

1. 运行时常量池
2. 含义：（将符号引用转化为直接引用）

一个类加载到 JVM 中后对应一个运行时常量池，运行时常量池相对于 Class 文件常量池来说具备动态性。

Class 文件常量池只是一个静态存储结构，里面的引用都是符号引用。

运行时常量池可以在运行期间将符号引用解析为直接引用，具有动态性。可以说运行时常量池就是用来索引和查找字段和方法名称和描述符的。

给定任意一个方法或字段的索引，通过这个索引最终可得到该方法或字段所属的类型信息和名称及描述符信息，这涉及到方法的调用和字段获取。

1. 内容：

直接引用

1. 位置：

方法区。

1. 数量：

每个class一个，即每个类一个。