

String 与 JVM

参考资料：

https://blog.csdn.net/qq_40301026/article/details/9498983

https://blog.csdn.net/clam_clam/article/details/6831345

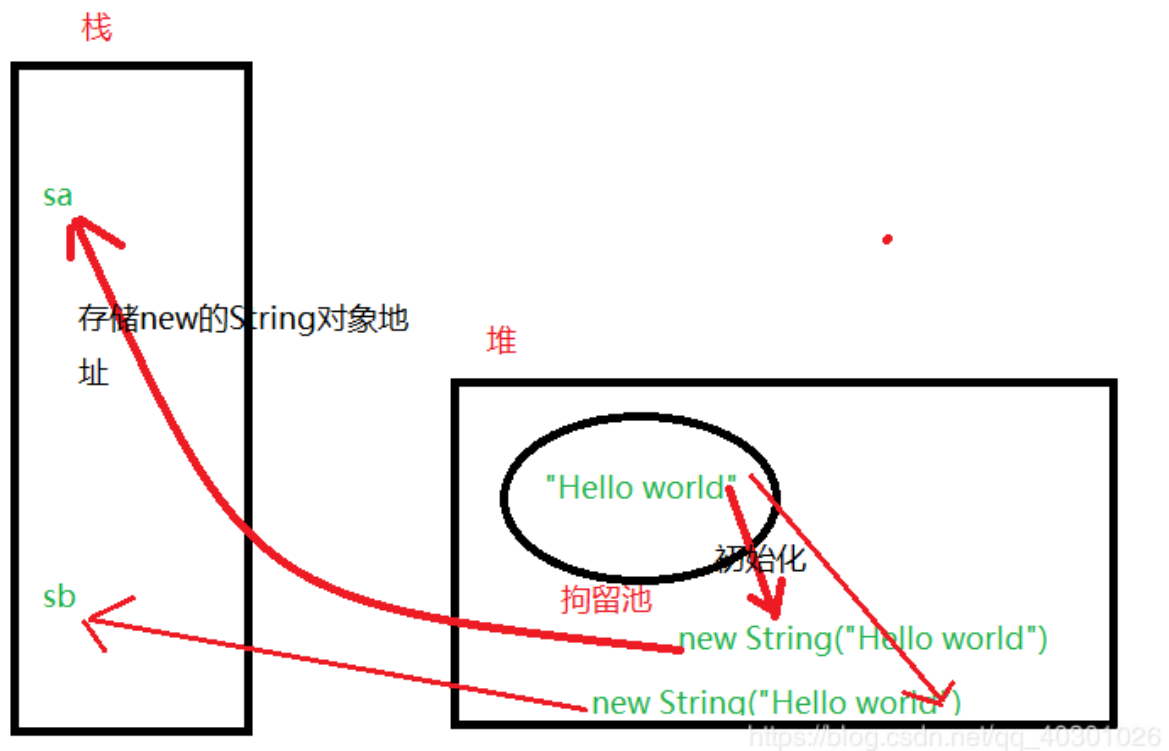
拘留池(intern pool)

- 公共语言运行库会自动维护一个名为 **拘留池(intern pool)** 的表，它包含在程序中声明的每个唯一字符串常数的单个实例，以及以编程方式添加的 String 的任何唯一实例。
- 拘留池表数据在 **程序编译时** 完成，即编译时就往拘留池内存入原来池里没有的字符串。

代码解释

case1

```
String sa = new String("Hello world");  
String sb = new String("Hello world");  
System.out.println(sa==sb); //false
```

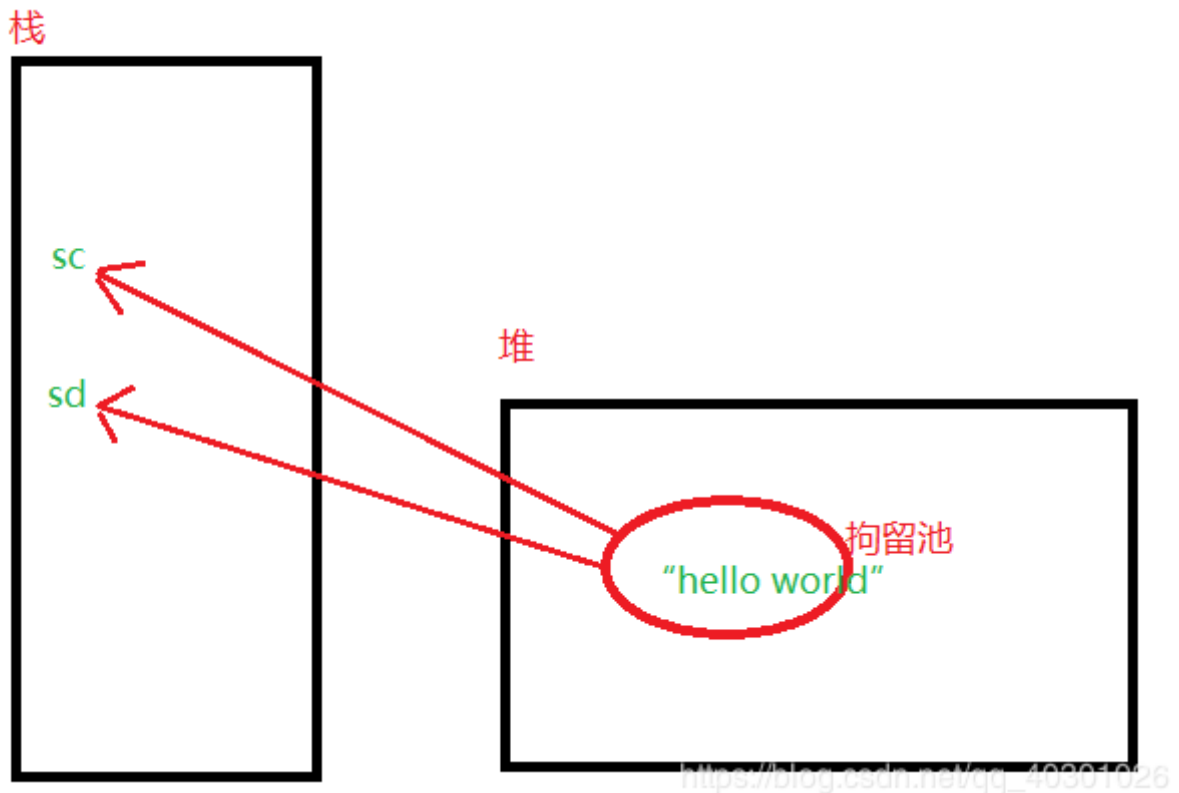


- 程序编译时拘留池中存入“Hello world”;

- new String 对象时只是利用拘留池的字符串去构造对象。所以最后sa和sb分别指向地址不同的两个对象。

case2

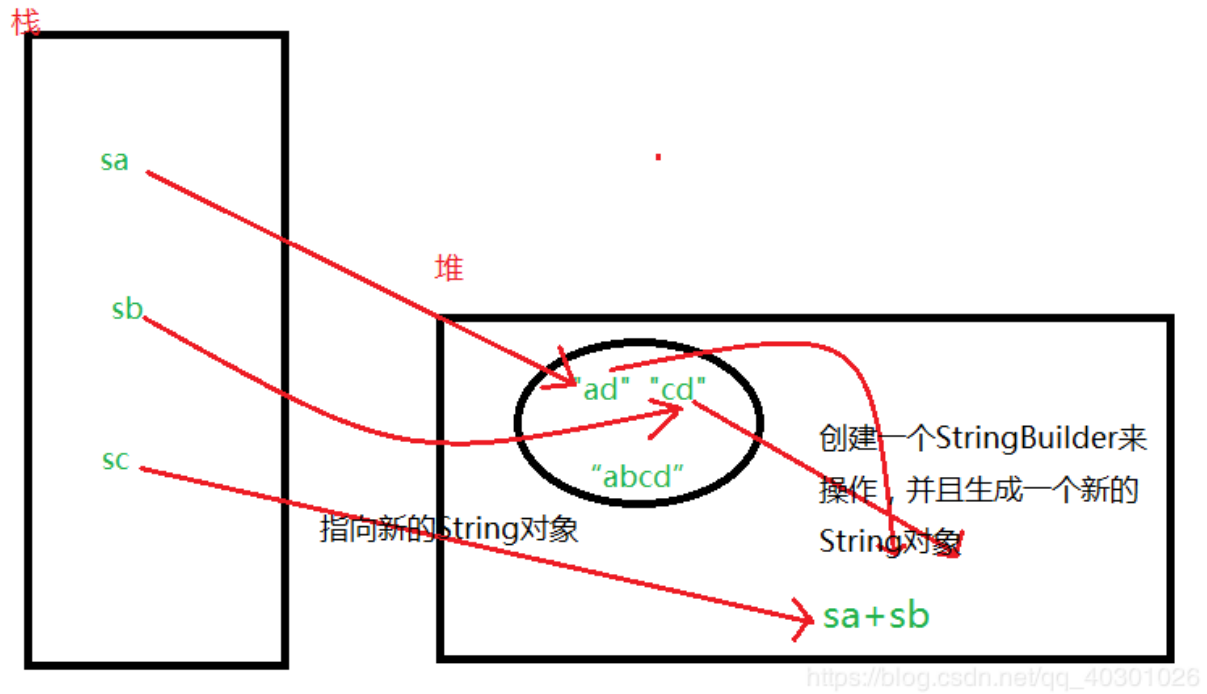
```
String sc = "Hello world";  
String sd = "Hello world";  
System.out.println(sc==sd);//true
```



- 程序编译时拘留池中存入“Hello world”，拘留池中之存入唯一字符串；
- sc和sd分别指向同一个拘留池中字符串地址。

case3

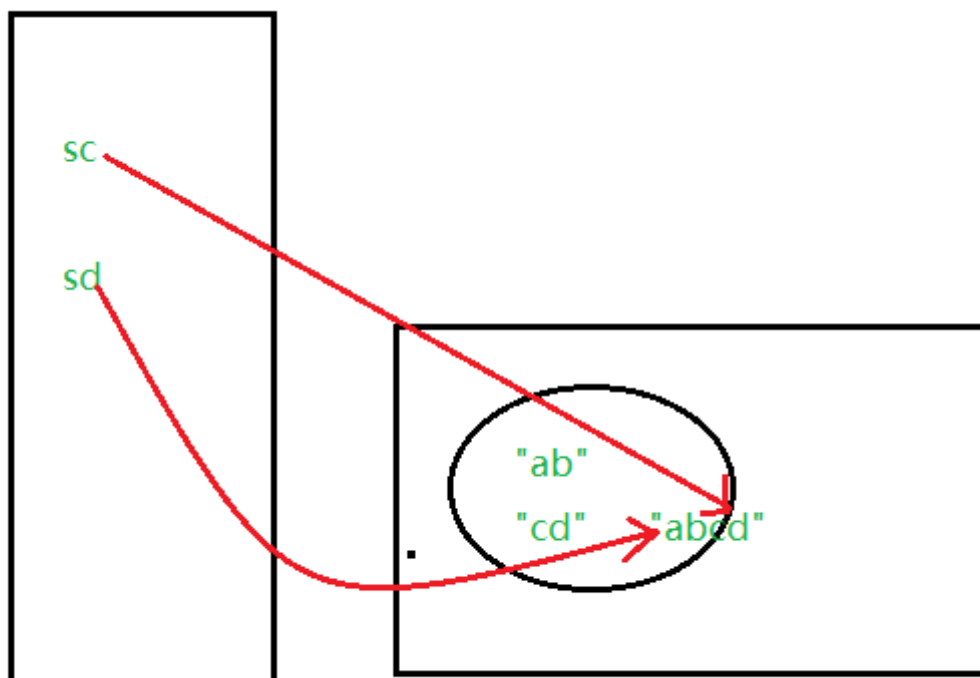
```
String sa = "ab";  
String sb = "cd";  
String sc = sa + sb;  
String s = "abcd";  
System.out.println(sc==s);//false
```



- 拘留池内存有三个字符串：“ad”，“cd”，“abcd”；
- sc是在运行时算出指向一个新的string对象。
(注：String是 `final` 类，每次计算都是创建一个 新的String对象)

case4

```
String sc="ab"+"cd";
String sd="abcd";
System.out.println(sc==sd);//true
```



- “ab” + “cd” 会在编译时就合成常量“abcd”存入拘留池。

