String 类：

String 类是有字符串构成的序列。在Java的源码中，String 类采用的依旧是和其他语言相同的 Char 类型数组为基础来建立对应的 String 类数值域。但是从源代码中我们可以发现它的数值域采用的均是 final 标识符，因此 Java 中的 String 类一旦赋值之后就不能够再对其中的数据进行修改。

分析它的数值域我们可以看到以下一些数值：

|  |  |
| --- | --- |
| final char[] value; | 字符数组 |
| final int count; | 字符数组长度 |
| private int cachedHashCode; | 哈希基准码 |
| final int offset; | 起始偏移 |

在准备层面，String 还设置了 CaseInsensitiveComparator 用于创建私有的比较器来存储之后相关的比较结果。

String 包含了常规的无参构造函数、复制构造函数、C风格构造函数、8字节ascii形式构造函数、16 位 Unicode Byte类型解码构造、支持和 StringBuffer 类的线程间同步复制构造、StringBuilder 类的复制构造和是否拷贝构造。

其中最后一个构造中，如果选择非拷贝构造，则直接采用引用构造，即将新构造的 String 直接用对象 String 的引用来赋值，仅仅改变 offset 值。否则，则将会新建一个 Char 数组来保存新产生的 String 类的数据。

常规的成员函数包括了对其相关数值域的访问取值、字符串截取并保存成不同类型的数组等。

String 类对于 equals 比较区别通过 instanceof 将其本类比较和 StringBuffer 类以及 StringBuilder 类区分开来。由于 String 数值域的不可变性，本类互相比较优先比较引用地址来加速比较，而和其他的比较则采用常规的优先比较长度其次比较内容的算法。同时，比较算法也包括了上述说到的无视大小写的比较。

String 的比较采用的是按字符扫描，返回字符比较值。

String 类提供了相关的子串操作，包括子串的判断、开头结尾的判断、字符的查找、子串的查找、子串的截取和获得、正则表达式的匹配以及替换、字符串的分裂、大小写转换以及和其他数值类型的转换。

String 中包含了其 hash 码的生成，通过将每一位字符的值乘以 31 之后再求和来获得。

对于 String 类的修改操作，由于 String 的不可变性，其所有的操作最终都会生成新的 String 对象。链接的时候只有空串才会对原 String 对象没有任何影响，而单个字符的替换就需要对整个 String 进行全部的重构，时间消耗非常大，但是空间上来说，它在结束原来的元素调用之后就会释放原来的空间，因此空间开销在可以接受的范围内。

String 类通过常量池来维护整个 String 类的相同字符串之间的地址赋值，能够比较方便的直接赋值引用，但是在当常量池较为庞大的时候，寻找所花费的时间也随之增大。

StringBuffer 类：

在参照 String 代码的基础上，我们可以发现其整体的成员函数结构等和 String 没有太大的差别，只是在基础数值域方面产生了比较大的差别。

StringBuffer 类和 String 类最大的区别就是，StringBuffer 类的数值域没有使用 final 来修饰，也就是说，我们可以对其数值域进行操作，而不是像 String 类一样，对其进行一次赋值之后就不能够在对其进行改变。所以我们可以认为，StringBuffer 是一个可以进行数值域修改的 String 类。对应 String 类的数值域，在其基础上，StringBuffer 提供了对其数值域进行修改的相关 set 函数。同时对于其内部的 value 域，也提供了相关的函数接口来修改其中的值。

这些接口包括，对指定字符的创建、插入、替换和删除、在末尾扩展其他类型的数据并转化成字符串类型、以及和 String 类以及 StringBuilder 类的转换。此外，它还可以通过调用 String 的相关操作函数来实现其中的一些功能。

StringBuffer 与 StringBuilder 最大的区别就在于，StringBuffer 的所有成员函数都保证了线程的同步，而 StringBuilder 则没有保证。

StringBuilder 类：

同 StringBuffer ，速度稍微低于 StringBuffer。

共同点：

通过三者的源代码我们可以看到它们均链接了CharSequence 接口，即三者均实现了 CharSequence 接口中的内容，三者存在共同的函数接口，方便三者之间的相互转化和调用。

StringBuffer 类和 StringBuilder 类两者基本相同，区别就是 Stringbuilder 是非线程安全的而 StringBuffer 是线程安全的。

一些总结：

经过测试，StringBuilder 类在单线程下速度最快，而 Stringbuffer 次之， String 最慢；但是，String 类的代码量最少，便于书写。

其中速度差主要来自于对于类的创建和修改上。对于 String 类，由于其不可修改的特性，因此无论是赋值操作，还是构造操作，都需要重新去申请新的地址，并开辟出固定大小的空间来，但是对于 StringBuffer 和 StringBuilder 来说，他们不需要去开辟新的空间，在O(1) 时间内就可以实现对其中的字符串的修改和维护操作，因此效率更高速度更快。

如果要操作少量的数据用String 类

单线程操作字符串缓冲区下操作大量数据 StringBuilder 类

多线程操作字符串缓冲区下操作大量数据 StringBuffer 类