https://mp.weixin.qq.com/s/FuRXhaZaTw65fhopu8KBLQ

# Java 中的 String 有没有长度限制？

## String的长度限制

想要搞清楚这个问题，首先我们需要翻阅一下String的源码，看下其中是否有关于长度的限制或者定义。

String类中有很多重载的构造函数，其中有几个是支持用户传入length来执行长度的：



可以看到，这里面的参数length是使用int类型定义的，那么也就是说，String定义的时候，最大支持的长度就是int的最大范围值。

根据Integer类的定义，java.lang.Integer#MAX\_VALUE的最大值是2^31 - 1;

那么，我们是不是就可以认为String能支持的最大长度就是这个值了呢？

其实并不是，这个值只是在运行期，我们构造String的时候可以支持的一个最大长度，而实际上，在运行期，定义字符串的时候也是有长度限制的。

如以下代码：



当我们使用如上形式定义一个字符串的时候，当我们执行javac编译时，是会抛出异常的，提示如下：



形式定义的时候无法编译呢？其实，形如String s = "xxx";定义String的时候，xxx被我们称之为字面量，这种字面量在编译之后会以常量的形式进入到Class常量池。那么问题就来了，因为要进入常量池，就要遵守常量池的有关规定。

## 常量池限制

我们知道，javac是将Java文件编译成class文件的一个命令，那么在Class文件生成过程中，就需要遵守一定的格式。

根据《Java虚拟机规范》中第4.4章节常量池的定义，CONSTANT\_String\_info 用于表示 java.lang.String 类型的常量对象，格式如下：



其中，string\_index 项的值必须是对常量池的有效索引， 常量池在该索引处的项必须是 CONSTANT\_Utf8\_info 结构，表示一组 Unicode 码点序列，这组 Unicode 码点序列最终会被初始化为一个 String 对象。

CONSTANT\_Utf8\_info 结构用于表示字符串常量的值：



其中，length则指明了 bytes[]数组的长度，其类型为u2，

通过翻阅《规范》，我们可以获悉。u2表示两个字节的无符号数，那么1个字节有8位，2个字节就有16位。

16位无符号数可表示的最大值位2^16 - 1 = 65535。

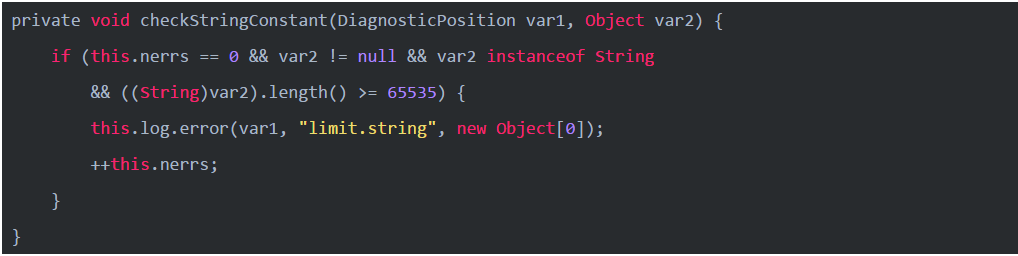
也就是说，Class文件中常量池的格式规定了，其字符串常量的长度不能超过65535。

那么，我们尝试使用以下方式定义字符串：



尝试使用javac编译，同样会得到"错误: 常量字符串过长"，那么原因是什么呢？

其实，这个原因在javac的代码中是可以找到的，在Gen类中有如下代码：

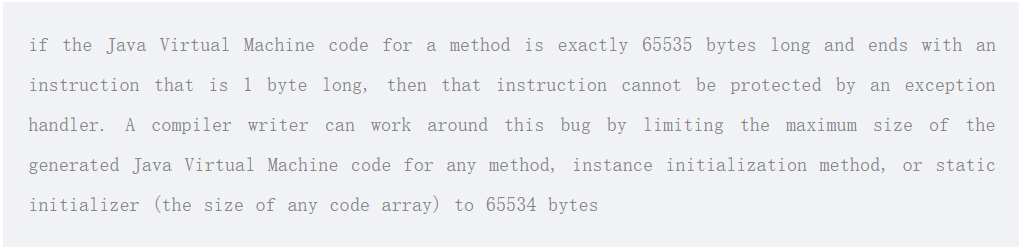


代码中可以看出，当参数类型为String，并且长度大于等于65535的时候，就会导致编译失败。

这个地方大家可以尝试着debug一下javac的编译过程（视频中有对java的编译过程进行debug的方法），也可以发现这个地方会报错。

如果我们尝试以65534个字符定义字符串，则会发现可以正常编译。

其实，关于这个值，在《Java虚拟机规范》也有过说明：

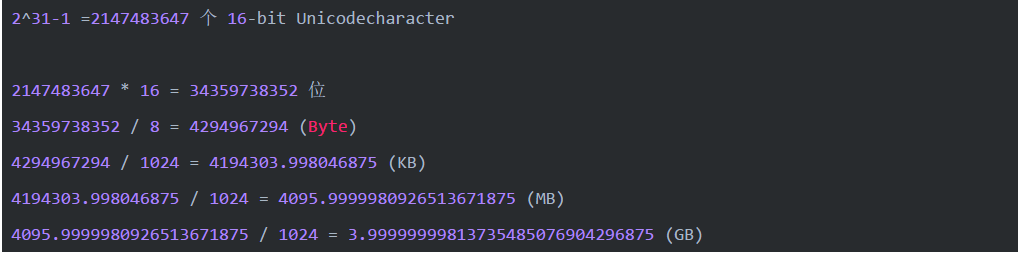


## 运行期限制

上面提到的这种String长度的限制是编译期的限制，也就是使用String s= “”;这种字面值方式定义的时候才会有的限制。

那么，String在运行期有没有限制呢，答案是有的，就是我们前文提到的那个Integer.MAX\_VALUE ，这个值约等于4G，在运行期，如果String的长度超过这个范围，就可能会抛出异常。(在jdk 1.9之前）

int 是一个 32 位变量类型，取正数部分来算的话，他们最长可以有



有近 4G 的容量。

很多人会有疑惑，编译的时候最大长度都要求小于65535了，运行期怎么会出现大于65535的情况呢。这其实很常见，如以下代码：



到的字符串长度就有10万，另外我之前在实际应用中遇到过这个问题。

之前一次系统对接，需要传输高清图片，约定的传输方式是对方将图片转成BASE6编码，我们接收到之后再转成图片。

在将BASE64编码后的内容赋值给字符串的时候就抛了异常。

## 总结

字符串有长度限制，在编译期，要求字符串常量池中的常量不能超过65535，并且在javac执行过程中控制了最大值为65534。

在运行期，长度不能超过Int的范围，否则会抛异常。