



链滴

提问：以下 Java 代码如何精简，或用其他语言用尽可能少的代码来实现

作者：[iTanken](#)

原文链接：<https://ld246.com/article/1516849288501>

来源网站：[链滴](#)

许可协议：[署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# 1. 功能要求

实现传入一个字符串，指定每行的长度，返回换行后的字符串（可忽略参数 endStr，特殊尾行不处换行）。传入的参数可能是任意的，代码中需防止出现空指针和下标越界等异常。

可以直接精简此 Java 代码或使用 C、Go、Python 等你喜欢的任何一种编程语言。

因为自己用 Java 实现后感觉实在是太啰嗦了，故有此问。

# 2. Java 代码

```
/**
 * 处理字符串自动换行
 * @param lineLength 每行长度
 * @param inStr 待转换字符串
 * @param lineSepar 换行符
 * @param endStr 最后一行
 * @return 处理后的字符串
 */
public static String wordWrapData(int lineLength, String inStr, String lineSepar, String endStr) {
    if (inStr == null && endStr == null) {
        return "";
    }
    lineSepar = lineSepar == null ? System.getProperty("line.separator") : lineSepar;
    StringBuilder outStr = new StringBuilder(128);
    boolean enable = inStr != null && inStr.length() > lineLength && lineLength > 0;
    int i = 0, newLength = lineLength, length = (inStr == null ? 0 : inStr.length());
    for (; enable && i < length - lineLength; newLength += lineLength, i += lineLength) {
        outStr.append(inStr.substring(i, newLength)).append(lineSepar);
        if (newLength < length && (newLength + lineLength) > length) {
            outStr.append(inStr.substring(newLength, length));
        }
    }
    if (!enable && inStr != null) {
        outStr.append(inStr);
    }
    if (endStr != null) {
        outStr.append(lineSepar).append(endStr);
    }
    return outStr.toString();
}

/**
 * TEST
 * @param args
 */
public static void main(String[] args) {
    String test = "12345678901234567890123456789012345678901234567890";
    System.out.println(wordWrapData(test, "\r\n", "", 18));
    System.out.println(wordWrapData(test, null, null, 0));
    System.out.println(wordWrapData(null, null, null, 18));
    System.out.println(wordWrapData("只有第一行", null, null, 18));
}
```

```
    System.out.println(wordWrapData(null, null, "只有最后一行", 18));
    System.out.println(wordWrapData("第一行", null, "最后一行", 18));
}
```

### 3. 执行结果

#### 第一次调用:

```
123456789012345678
901234567890123456
7890
```

#### 第二次调用:

```
1234567890123456789012345678901234567890
```

#### 第三次调用:

#### 第四次调用:

```
只有第一行
```

#### 第五次调用:

```
只有最后一行
```

#### 第六次调用:

```
第一行
最后一行
```

<p class="hidden"> PS. 在 hacpai 发表提示存在保留词, 只能用 Solo 同步过来了 :joy: </p>