

如何应用Java的可变参数Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/620/2021_2022__E5_A6_82_E4_BD_95_E5_BA_94_E7_c104_620996.htm

可变参数 Java语言在JDK1.5中首次推出可变参数，variable arguments，或简称varargs。这一新语言特征给软件开发人员在编写方法重载时提供了方便和灵活性。但可变参数的应用并不像想象的那么简单，使用时有其特殊要求和局限性。重载的最好例子读者朋友可能有这样的编程经历：在编写一个方法时，其参数随着程序运行的条件而变化，在编译期间无法确定。具体地讲，例如编写一个打印参加聚会party的程序，其中方法printInvitation()将根据作为参数的参加人姓名，打印邀请卡。但这个参数的数量事先并不确定。当然可以编写许多重载的方法来解决这个问题，如：
void printInvitation(String name).
void printInvitation(String name1, String name2).
void printInvitation(String name1, String name2, String name3). ... 问题是编写多少个重载的方法才可以解决给所有参加者打印邀请卡？也许需要改变您的程序设计，而使用数组或者链接表了。应用可变参数可以方便、灵活地解决这类问题。例如：
//完整程序存在本书配套资源目录Ch12名为VarargsTest.java
void printInvitation(String...names) { for (String name : names) {
makeCard(name). //调用方法按照姓名打印邀请卡
System.out.println("Recording info: invitation card has been printed for " name). } } 这里，(String...names)便是可变参数。它包括从0到任意个相同类型的参数。在编译期间，这个可变参数将被转换为字符串数组形式，即：
void printInvitation(String[]

names) 如下是调用这个方法的例子：`printInvitation("李刚", "David Smith"). printInvitation("Greg Wu", "Paul Nguyen", "Liu Wei", "张新"); printInvitation().` //无参数 当在无参数情况下调用这个方法时，将不执行任何这个方法中的代码。 如下是运行结果：
Recording info: invitation card has been printed for 李刚
Recording info: invitation card has been printed for David Smith
Recording info: invitation card has been printed for Greg Wu
Recording info: invitation card has been printed for Paul Nguyen
Recording info: invitation card has been printed for Liu Wei
Recording info: invitation card has been printed for 张新

怎样工作 可变参数也不神秘。实际上，JVM将根据程序中调用这个方法时提供的参数数量，来装载和运行它。 可变参数的简单语法格式为：`methodName([argumentList], dataType...argumentName)`。 其中：`argumentList`普通参数，可选项。`dataType`数据类型或者类。自动转换成`dataType`代表的数组。`...`Java的操作符。表示0到多个。必须是3个点。

`argumentName`参数名。 注意，可变参数必须在最后。 下面是应用可变参数的更多例子：`// 完整程序存在本书配套资源目录Ch12名为VarargsTest.java`

```
public static int  
sumInts(int...numbers) { //可变整数数组类型参数 int sum = 0.  
for (int num : numbers) sum =num. return sum. } 100Test
```

下载频道开通，各类考试题目直接下载。详细请访问

www.100test.com