

JAVA认证:创建InetAddress对象的四个静态方法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\\_ti2020/559/2021\\_2022\\_JAVA\\_E8\\_A](https://www.100test.com/kao_ti2020/559/2021_2022_JAVA_E8_A)

E\_A4\_E8\_AF\_81\_c104\_559560.htm InetAddress类是Java中用于描述IP地址的类。它在java.net包中。在Java中分别用Inet4Address和Inet6Address类来描述IPv4和IPv6的地址。这两个类都是InetAddress的子类。由于InetAddress没有public的构造方法，因此，要想创建InetAddress对象，必须得依靠它的四个静态方法。InetAddress可以通过getLocalHost方法得到本机的InetAddress对象，也可以通过getByName

、getAllByName和getByAddress得到远程主机的InetAddress对象。一、getLocalHost方法使用getLocalHost可以得到描述本机IP的InetAddress对象。这个方法的定义如下：

```
public static  
InetAddress getLocalHost() throws UnknownHostException
```

这个方法抛出了一个UnknownHostException异常，因此，必须在调用这个方法的程序中捕捉或抛出这个异常。下面的代码演示了如何使用getLocalHost来得到本机的IP和计算机名。

```
package inet. import java.net.*. public class MyInetAddress1 { public  
static void main(String[] args) throws Exception { InetAddress  
localAddress = InetAddress.getLocalHost().
```

```
System.out.println(localAddress).} } 运行结果：
```

ComputerName/192.168.18.10 在InetAddress类中覆盖了Object类的toString方法，实现如下：

```
public String toString() { return  
((hostName != null) ? hostName : "") "/" getHostAddress().}
```

从上面的代码可以看出，InetAddress方法中的toString方法返回了用“/”隔开的主机名和IP地址。因此，在上面的代码中直接

通过 `localAddress` 对象来输出本机计算机名和IP地址（将对象参数传入 `println` 方法后，`println` 方法会调用对象参数的 `toString` 方法来输出结果）。当本机绑定了多个IP时，`getLocalHost` 只返回第一个IP。如果想返回本机全部的IP，可以使用

`getAllByName` 方法。二、`getByName` 方法 这个方法 是 `InetAddress` 类最常用的方法。它可以通过指定域名从DNS中得到相应的IP地址。`getByName` 一个 `String` 类型参数，可以通过这个参数指定远程主机的域名，它的定义如下：`public static InetAddress getByName(String host) throws`

`UnknownHostException` 如果 `host` 所指的域名对应多个IP

，`getByName` 返回第一个IP。如果本机名已知，可以使用 `getByName` 方法来代替 `getLocalHost`。当 `host` 的值是 `localhost` 时，返回的IP一般是 `127.0.0.1`。如果 `host` 是不存在的域名

，`getByName` 将抛出 `UnknownHostException` 异常，如果 `host` 是IP地址，无论这个IP地址是否存在，`getByName` 方法都会返回这个IP地址（因此 `getByName` 并不验证IP地址的正确性）。

下面代码演示了如何使用 `getByName` 方法：`package inet.`

```
import java.net.*; public class MyInetAddress2 { public static void main(String[] args) throws Exception { if (args.length == 0) return; String host = args[0]; InetAddress address = InetAddress.getByName(host); System.out.println(address); } }
```

测试1：远程主机 `www.csdn.net` 执行如下命令：`java`

`inet.MyInetAddress2 www.csdn.net` 运行结果：

`www.csdn.net/211.100.26.124` 测试2：本机名 `ComputerName` 执行如下命令：

`java inet.MyInetAddress2 ComputerName` 运行结果：

`ComputerName/192.168.18.10` 测试3：代表本机的 `localhost`

执行如下命令：`java inet.MyInetAddress2 localhost` 运行结果：`localhost/127.0.0.1` 对于本机来说，除了可以使用本机名或localhost外，还可以在hosts文件中对本机做“IP/域名”映射（在Windows操作系统下）。这个文件在C:\WINDOWS\system32\drivers\etc中。打开这两个文件中，在最后加一行如下所示的字符串：`192.168.18.100 www.mysite.com`

测试4：本机域名www.mysite.com 执行如下命令：`java inet.MyInetAddress2 www.mysite.com` 运行结果：`www.mysite.com/192.168.18.100`

getName方法除了可以使用域名作为参数外，也可以直接使用IP地址作为参数。如果使用IP地址作为参数，输出InetAddress对象时域名为空（除非调用getHostName方法后，再输出InetAddress对象。getHostName方法将在下面的内容介绍）。读者可以使用129.42.58.212作为MyInetAddress2的命令行参数（这是www.ibm.com的IP），看看会得到什么结果。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](http://www.100test.com)