

RMI规范--第四章客户机接口 PDF转换可能丢失图片或格式，
建议阅读原文

https://www.100test.com/kao_ti2020/145/2021_2022_RMI_E8_A7_84_E8_8C_83-_c104_145021.htm 程序员在编写使用远程对象的

applet 或应用程序时，需要注意 java.rmi 包中可用的 RMI 系统
客户机可视接口。 4.1 远程接口 package java.rmi. public

```
interface Remote {} java.rmi.Remote
```

 接口用来识别所有远程接口

；所有远程对象必须直接或间接实现此接口。实现类可以实现任意数目的远程接口，并可扩展其它远程实现类。RMI 提供一些远程对象实现可以扩展的类，有助于远程对象的创建。

这些类是 java.rmi.server.UnicastRemoteObject

和 java.rmi.activation.Activatable。有关如何定义远程接口的详细信息，参见“java.rmi.Remote 接口”（2.4.1）一节。 4.2

RemoteException 类类 java.rmi.RemoteException 是许多在执行远程方法调用时可能发生的、与通信有关的异常的通用超类。

远程接口中的每种方法（也是一个接口）必须在其 throws 子句中列出 RemoteException（或其超类，

如 java.io.IOException 或 java.lang.Exception）。 package

```
java.rmi. public class RemoteException extends java.io.IOException
```

```
{ public Throwable detail. public RemoteException(). public
```

```
RemoteException(String s). public RemoteException(String s,
```

```
Throwable ex). public String getMessage(). public void
```

```
printStackTrace(). public void printStackTrace(java.io.PrintStream
```

```
ps). public void printStackTrace(java.io.PrintWriter pw). }
```

RemoteException 可用详细消息（即 s）和一个嵌套异常（即

ex，Throwable）进行构造。嵌套异常 ex 在构造函数的第三种

形式中被指定为参数，通常是 RMI 调用过程中发生的基本 I/O 异常。getMessage 方法返回异常的详细消息，包括嵌套异常（如果有）中的消息。printStackTrace 方法在类 java.lang.Throwable 中将被覆盖掉，以打印嵌套异常的堆栈跟踪。

4.3 Naming 类

java.rmi.Naming 类提供存储和获得对远程对象注册服务程序中远程对象进行引用的方法。Naming 类中的方法以如下形式的，URL 格式的 java.lang.String 作为其中的一个参数：`//host:port/name` 其中 host 是注册服务程序所在的主机（远程或本地），port 是注册服务程序接收调用的端口号，name 是注册表未作出解释的简单字符串。host 和 port 是可选的。如果省略了 host，则主机缺省值为本地主机。如果省略了 port，则端口缺省值为 1099，即 RMI 系统注册服务程序 rmiregistry 所用的“众所周知”的端口。为远程对象绑定名称即为稍后使用的远程对象关联或注册名称，可用于查询该远程对象。可以使用 Naming 类的 bind 或 rebind 方法将远程对象与名称相关联。当远程对象已用 RMI 注册服务程序在本地主机上进行过注册（绑定）后，远程（或本地）主机上的调用程序就可以按名称查询远程对象、获得其引用，然后在对象上调用远程方法。必要时，某一主机上运行的服务器可以共享一个注册服务程序。服务器的各个进程也可创建和使用自己的注册服务程序（详细信息，参见 java.rmi.registry.LocateRegistry.createRegistry 方法）。

```
package java.rmi; public final class Naming { public static Remote lookup(String url) throws NotBoundException, java.net.MalformedURLException, RemoteException. public static void bind(String url, Remote obj) throws AlreadyBoundException,
```

```
java.net.MalformedURLException,RemoteException. public static
void rebind(String url, Remote obj) throws RemoteException,
java.net.MalformedURLException. public static void unbind(String
url) throws RemoteException, NotBoundException,
java.net.MalformedURLException. public static String[] list(String
url) throws RemoteException, java.net.MalformedURLException. }
lookup 方法返回与名称的文件部分相关联的远程对象。如果
名称未绑定到对象上，则抛出 NotBoundException。 bind 方法
将把指定名称绑定到远程对象上。如果该名称已绑定到某一
对象上，则抛出 AlreadyBoundException。 rebind 方法总将名
称绑定到对象上，无论该名称是否已绑定。原有绑定将丢失
。 unbind 方法将取消名称和远程对象间的绑定。如果没有绑
定，则抛出 NotBound Exception。 list 方法返回一个 String 对
象的数组，该对象包含注册服务程序中绑定 URL 的快照。为
了向注册服务程序查询其内容列表，只需要 URL 上的主机名
和端口信息；因，URL 的“file”部分将被忽略。
```

注意 - 这些方法也可能抛出 java.rmi.AccessException
。AccessException 表示调用程序无执行特定操作的权限。例
如，只有运行注册服务程序的主机上的本地客户机才允许执
行 bind、rebind 和 unbind 操作。但任何非本地客户机都可调
用 lookup 操作。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