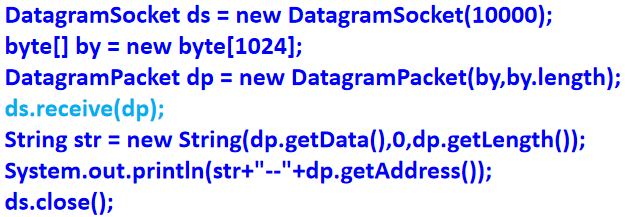
* **UDP网络通信**
* 类**DatagramSocket**和**DatagramPacket**实现了基于UDP协议网络程序。
* UDP数据报通过数据报套接字DatagramSocket发送和接收，**系统不保证UDP数据报一定能够安全送到目的地，也不能确定什么时候可以抵达**。
* DatagramPacket对象封装了UDP数据报，在数据报中包含了发送端的IP地址和端口号以及接收端的IP地址和端口号。
* UDP协议中每个数据报都给出了完整的地址信息，因此无须建立发送方和接收方的连接。
* **流程：**
* **DatagramSocket与DatagramPacket**；
* 建立发送端，接收端；
* 建立数据包；
* 调用Socket的发送、接收方法；
* 关闭Socket。
* **发送端与接收端是两个独立的运行程序**
* 发送端：



* 接收端：

在接收端，要指定监听的端口



*/\*\*  
 \* 发送端  
 \*/*@Test  
**public void** send() {  
 DatagramSocket ds = **null**;  
 **try** {  
 ds = **new** DatagramSocket();  
 **byte**[] b = **"你好，我是要发送的数据"**.getBytes();  
  
 *//创建一个数据报，每一个数据报不能大于64k，都记录着数据信息，  
 // 发送端的IP、端口号，以及要发送到的接收端的IP、端口号* DatagramPacket pack = **new** DatagramPacket(b, 0, b.**length**,  
 InetAddress.*getByName*(**"127.0.0.1"**), 9090);  
  
 *//调用DatagramSocket的发送方法* ds.send(pack);  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 } **finally** {  
 **if** (ds != **null**) {  
 *//关闭* ds.close();  
 }  
 }  
}  
  
*/\*\*  
 \* 接收端  
 \*/*@Test  
**public void** receive() {  
 DatagramSocket ds = **null**;  
 **try** {  
 ds = **new** DatagramSocket(9090);  
 **byte**[] b = **new byte**[1024];  
 DatagramPacket pack = **new** DatagramPacket(b, 0, b.**length**);  
 ds.receive(pack);  
  
 String str = **new** String(pack.getData(), 0, pack.getLength());  
 System.***out***.println(str);  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 } **finally** {  
 **if** (ds != **null**) {  
 ds.close();  
 }  
 }  
}