

```

/**
 * 客户端类
 * 监听 ZooKeeper 中的指定节点，若子节点状态发生变化，则获取子节点列表
 */
public class PCClient {
    //ZooKeeper 连接字符串
    private static final String CONNETC_STR =
"centos01:2181,centos02:2181,centos03:2181";
    //连接超时时间（两秒）
    private static final int SESSION_TIME_OUT = 2000;
    //指定需要监听的 ZooKeeper 中的节点，该节点需要提前在 ZooKeeper 中创建
    private static final String PARENT_NODE = "/serverGroup";
    ZooKeeper zk = null;
    //客户端连接 ZooKeeper
    private void connectZookeeper() throws Exception {
        zk = new ZooKeeper(CONNETC_STR, SESSION_TIME_OUT, new Watcher()) {①
            //监听事件的回调方法
            public void process(WatchedEvent event) {②
                //若指定节点下的子节点状态发生变化，重新获取服务器列表，并重新注册监听
                if(event.getType() == Event.EventType.NodeChildrenChanged &&
(PARENT_NODE).equals(event.getPath())){
                    try {
                        //获取服务器列表
                        getServerList();
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                }
            }
        });
        //获取在线服务器列表
        getServerList();
    }

/**
 * 获取在线服务器列表
 */
private void getServerList() throws Exception {
    //获取服务器子节点列表，并重新对父节点进行监听。参数 true 代表设置监听
    List<String> subNodeList = zk.getChildren(PARENT_NODE, true);③
    //先创建一个 List 集合存储服务器信息
    List<String> newServerList = new ArrayList<String>();
    //循环服务器子节点列表，并将每个子节点的元数据字符串添加到 newServerList 集合
    for (String subNodeName : subNodeList) {
        byte[] data = zk.getData(PARENT_NODE+"/"+subNodeName, false, null);
        //输出节点元数据字符串
        newServerList.add(new String(data));
    }
    //输出当前在线服务器列表
    if(newServerList==null||newServerList.size()==0)

```