

Chapter 14 网络编程

Key Point:

- 基本的 Socket 编程
- 多线程的 TCP 服务器

答案:

1. TCP 和 UDP 都是传输层协议，TCP 是有连接（有连接|| 无连接）的协议，UDP 是无连接（有连接|| 无连接）的协议。这两种协议中，TCP 协议更安全，而 UDP 协议传输效率更高。

2. ABC

3. AB

ServerSocket 中没有 `getInputStream` 和 `getOutputStream` 方法

4. 填空

一般而言，创建一个 Tcp 客户端，有以下几步：

- 1) 创建一个 Socket 对象
- 2) 调用 `getInputStream` 方法和 `getOutputStream` 方法获得输入输出流
- 3) 利用输入输出流，读写数据
- 4) 关闭 socket

创建一个多线程的 Tcp 服务器，有以下几步：

- 1) 创建 `ServerSocket` 对象
- 2) 调用该对象的 `accept` 方法，以获取客户端的连接。该方法返回一个 `Socket` 对象。
- 3) 利用返回的对象，创建一个新线程

4) 在新线程中完成读写操作

5) 在新线程中调用 Socket 对象的 close 方法

5. (TCP 编程) 根据提示, 把下面的代码补充完整

```
//Client.java
```

```
import java.net.*;
```

```
import java.io.*;
```

```
public class Client{
```

```
    public static void main(String args[])throws Exception{
```

```
        Socket s;
```

```
        // 创建一个到“ 127.0.0.1: 9000 ”的 Tcp 连接
```

```
        s = new Socket( “ 127.0.0.1 ” , 9000);
```

```
        // 向 Tcp 连接输出“ Hello World ”并换行
```

```
        PrintWriter pw = new PrintWriter(s.getOutputStream());
```

```
        pw.println( “ Hello World ” );
```

```
        // 从服务器端读入一行文本, 并打印出来
```

```
        BufferedReader br = new BufferedReader(
```

```
            new InputStreamReader(s.getInputStream()));
```

```
        String str = br.readLine();
```

```
        System.out.println(str);
```

```
        s.close();
```

```
    }
```

```
}
```

```
//Server.java
```

```
import java.io.*;
```

```
import java.net.*;
```

```
public class Server{
```

```
    public static void main(String args[]) throws Exception {
```

```
        // 创建一个服务器端口对象
```

```
ServerSocket ss = new ServerSocket(9000);

// 获得一个客户的连接

Socket s = ss.accept();

// 读入一行文本

BufferedReader br = new BufferedReader(
    new InputStreamReader(s.getInputStream()));

String str = br.readLine();

// 在读入的文本后面加上+ + “ From Server ”

str += “ From Server ”;;

// 把处理之后的文本向客户端输出并换行

PrintWriter pw = new PrintWriter(s.getOutputStream());

pw.println( “ Hello World ” );

// 关闭连接

s.close();

}

}
```

6. ** (多线程, TCP)

参考 UpperCaseClient.java 和 UpperCaseServer.java

7. ** (多线程 TCP)

参考 FtpServer.java 和 FtpClient.java