**每日作业卷**

**Java基础第14天**

**UDP、TCP**

# 关卡1

## 训练案例1

### 训练描述：【讲义：1.1—ip地址和端口号】

1. 请写出IP地址的概念：
2. 请写出端口号的概念：

**答案：参见讲义1.1 ip地址和端口号**

### 操作步骤描述

（参考讲义1.1）

## 训练案例2

## **2.1训练描述：【讲义1.1—ip地址和端口号】**

1. 在计算机中,端口号用\_\_\_个字节,也就是16位的二进制表示,它的取值范围是\_\_\_\_。

**答案：参见讲义1.1 ip地址和端口号**

### 2.2操作步骤描述

（参考讲义1.1）

## 训练案例3

### 训练描述：【讲义1.2—InetAddress】

1. 在JDK中,IP地址使用\_\_\_\_类来表示,该类提供了许多和IP地址相关的操作.

**答案：参见讲义1.2-InetAddress**

### 操作步骤描述

（参考讲义1.2）

## 训练案例4

### 训练描述：【讲义1.2—InetAddress】

1. 创建新项目，按以下要求编写代码：
2. 使用InetAddress类获取本地计算机的IP地址和主机名
3. 使用InetAddress类获取百度(www.baidu.cn)主机的IP地址：

提示 getLocalHost()获取本机地址

getByName(“”)获取指定网络

**答案：见附件项目目录：day14作业\_Test1\_4**

### 操作步骤描述

（参考讲义1.2）

## 训练案例5

### 训练描述：【讲义2.1—UDP协议】

1. 判断下列说法是否正确

由于UDP面向无连接的协议,可以保证数据完整性,因此在传输重要数据时采用UDP协议;

**答案：**

**判断错误,因为面向无连接,容易丢失包,所以不能保证数据完整.**

### 操作步骤描述

（参考讲义2.1）

## 训练案例6

### 训练描述：【讲义2.2—TCP协议】

1. TCP协议中”三次握手”,第一次握手指的是什么：
2. 客户端再次向服务器发送确认信息,确认连接
3. 服务器向客户端回送一个响应,通知客户端收到了连接请求
4. 客户端向服务器发送请求,等待服务器确认
5. 以上答案全部错误

**答案：C**

### 操作步骤描述

（参考讲义2.2）

## 训练案例7

### 训练描述：【讲义3.1—DatagramPacket】

1. 使用UDP协议通信时,需要使用哪个类把要发送的数据打包?：
2. Socket
3. DatagramSocket
4. DatagramPacket
5. ServerSocket

**答案：C**

### 操作步骤描述

（参考讲义3.1）

## 训练案例8

### 训练描述：【讲义3.1—DatagramPacket】

1. 进行UDP协议通信时,在接收端若要获得发送端的IP地址,可以使用DatagramPacket的哪个方法?：
2. getAddress()
3. getPort()
4. getName()
5. getData()

**答案：A**

### 操作步骤描述

（参考讲义3.1）

## 训练案例9

### 训练描述：【讲义3.2—DatagramSocket】

1. 以下那个方法是DatagramSocket类用于发送数据的方法：
2. receive()
3. accept()
4. set()
5. send()

**答案：D**

### 操作步骤描述

（参考讲义3.2）

## 训练案例10

### 训练描述：【讲义3.3— UDP程序】

需求说明：创建新项目,通过UDP协议，完成UDP发送端，使用UDP给本机的接收端(8888端口)发送信息“Hello”

**答案：见附件项目目录：day14作业\_Test1\_10**

### 操作步骤描述

1. 创建项目：day14作业\_Test1\_10
2. 创建MainApp类，并包含main()方法；
3. 参考讲义1.2，按上述要求实现代码。

## 训练案例11

### 训练描述：【讲义4.1—ServerSocket】

1. 以下那个方法是ServerSocket类用于接收来自客户端请求：
2. receive()
3. accept()
4. set()
5. send()

**答案：B**

### 操作步骤描述

参考讲义4.1）

## 训练案例12

### 训练描述：【讲义4.3—简单的TCP网络程序】

1. 需求说明：创建新项目，按以下要求编写代码：

在项目下创建TCP 服务器端 端口号为8888

1:等待客户端连接 如果有客户端连接 获取到客户端对象

2:获取到客户端对象之后 当前在服务器读取数据客户端传送数据

**答案：见附件项目目录：day14作业\_Test1\_12**

### 操作步骤描述

1. 创建项目：day14作业\_Test1\_12
2. 创建TCPServer类，并包含main()方法；
3. 参考讲义4.3，按上述要求实现代码。

## 训练案例13

### 训练描述：【讲义4.3—简单的TCP网络程序】

1. 需求说明：创建新项目，按以下要求编写代码：

在项目下创建TCP 客户端

访问之前创建的服务器端,服务器端ip127.0.0.1 端口号8888

1:客户端连接服务器,并发送 hello.服务器,我是客户端.

2:开启上一题服务器,等待客户端连接,客户端连接并发送数据

**答案：见附件项目目录：day14作业\_Test1\_13**

### 操作步骤描述

1. 创建项目：day14作业\_Test1\_13创建TCPClient类，并包含main()方法；
2. 参考讲义4.3，按上述要求实现代码。

# 关卡2

## 训练案例1

### 训练描述：【UDP综合练习之网络聊天室】

1. 项目需求：请用户完成一个UDP接收端作为模拟网络聊天室的操作。

1:创建UDP传输下的聊天室发送端UDPSend类

2: 创建UDP传输下的聊天室接收端 Room类

**答案：见附件项目目录：day14作业\_Test2\_1**

### 操作步骤描述

1. 建立项目目录：day14作业\_Test2\_1
2. 创建UDPSend类，并包含main()方法；

在该main方法中

a、建立UDP的Socket服务；

b、将要发送的数据封装到数据包中；

c、通过UDP的Socket服务将数据包发送出去；

d、关闭Socket服务

3. 创建Room类 使用UDP接收端实现网络聊天效果

该类中写main方法 步骤如下

a、建立UDP的Socket服务，因为要接收数据，所以必须明确一个端口号；

添加while(ture)循环，不断的循环接收内容  
 b、创建数据包，用于存储接收到的数据，方便用数据包对象的方法解析这些数据；   
 c、使用UDP的Socket服务的receive方法接收数据并存储到数据包中；   
 d、通过数据包的方法解析这些数据； 

## 训练案例2

### 训练描述：【TCP程序】

1. 需求说明：使用TCP编写一个网络程序,设置服务器程序监听端口为8002,当于客户端建立后,向客户端发送”hello world”,客户端将信息输出

**答案：见附件项目目录：day14作业\_Test2\_2**

### 操作步骤描述

1. 建立项目：day14作业\_Test2\_2
2. 创建TCPServer类,包含main()方法

使用ServerSocket创建服务器对象,监听8002端口,调用accept()方法等待客户端连接,当于客户端连接上之后,获取输出流对象,输出”hello world”

1. 创建TCPClient类，包含main()方法

使用Socket创建客户端对象,指定服务器IP和端口号,与服务器端建立连接后,获取输入流对象,读取数据,并打印.

# 关卡3

## 训练案例1

### 训练描述：【TCP上传案例】

1. 需求说明：我们来做一个“文件上传案例”，有以下要求：

将项目中的一个图片,通过客户端上传至服务器

### 操作步骤描述

1,创建服务器，等待客户端连接

2,创建客户端Socket，连接服务器

3,获取Socket流中的输出流，功能：用来把数据写到服务器

4,创建字节输入流，功能：用来读取数据源(图片)的字节

5,把图片数据写到Socket的输出流中(把数据传给服务器)

6,客户端发送数据完毕，结束Socket输出流的写入操作，告知服务器端

7,获取Socket的输入流

8,创建目的地的字节输出流

9,把Socket输入流中的数据，写入目的地的字节输出流中

10,获取Socket的输出流, 作用：写反馈信息给客户端

11,写反馈信息给客户端

12,获取Socket的输入流 作用： 读反馈信息

13,读反馈信息