**每日作业卷**

**JavaSE第11天**

**网络编程**

传智播客.黑马程序员

# 基础题

## 练习一:ip地址和端口号概念

描述:

1. 请写出IP地址的概念：
2. 请写出端口号的概念：

答案:

IP地址:互联网协议地址(Internet Protocol Address),俗称IP.IP地址用来给一个网络中的计算机设备做唯一的编号.

端口号:端口号用来给计算机里的应用程序(进程)做唯一的标识,用2个字节表示的整数,取值范围0~65535.

## 练习二:UDP协议

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.

## 练习三:TCP协议

TCP协议中”三次握手”,第一次握手指的是什么：

答案

第一次握手:客户端向服务器发送请求,等待服务器确认

## 练习四:ServerSocket

以下哪个方法是ServerSocket类用于接收来自客户端请求：

1. receive()
2. accept()
3. set()
4. send()

答案

B

## 练习五:TCP网络程序

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

public class TCPServer {

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

  //1创建服务器对象

  ServerSocket  ss = new ServerSocket(8888);

  //2等待客户端连接   如果有客户端连接  获取到客户端对象

  Socket socket = ss.accept();

  //3当前在服务器中  要读取数据  需要输入流  流由谁提供 客户端

  InputStream in = socket.getInputStream();//获取输入流

  //4:读数据

  int len;

  byte[] buffer = new byte[1024];

  while((len=in.read(buffer))!=-1){

  System.out.println(new String(buffer, 0, len));

  }

  //释放资源

  in.close();

//   ss.close();服务器一般不会关闭

   }

}

## 练习六:TCP网络程序

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:

public class TCPClient {

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

//创建 Socket客户端对象

  Socket  socket = new Socket("127.0.0.1", 8888);

  //写数据  需要输出流  谁提供 客户端

  OutputStream out = socket.getOutputStream();

  //写数据

  out.write("hello.服务器,我是客户端.".getBytes());

  //释放资源

  out.close();

  socket.close();

  }

}

# 扩展题

## 练习一:TCP程序

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

操作步骤:

1. 建立项目
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.

代码:

public class TCPClient {

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

//创建 Socket客户端对象

  Socket  socket = new Socket("127.0.0.1", 8002);

  //读服务器数据

  InputStream in = socket.getInputStream();//获取输入流

  //读数据

  byte[] buffer = new byte[1024];

  int len = in.read(buffer);

  System.out.println(new String(buffer, 0, len));

  //释放资源

  in.close();

  socket.close();

  }

}

public class TCPServer {

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

  //1创建服务器对象

  ServerSocket  ss = new ServerSocket(8002);

  //2等待客户端连接   如果有客户端连接  获取到客户端对象

  Socket socket = ss.accept();

  //3当前在服务器中  将数据写到流中

  OutputStream out = socket.getOutputStream();

  out.write("hello world".getBytes());

  //释放资源

  out.close();

//   ss.close();服务器一般不会关闭

   }

}

## 练习二:TCP上传案例

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

答案

操作步骤:

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

代码:

public class TCPServer {

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

//1创建服务器对象

ServerSocket server = new ServerSocket(9898);

//2等待客户端连接   如果有客户端连接  获取到客户端对象

Socket s = server.accept();

//3当前在服务器中  读取数据

InputStream in = s.getInputStream();

//4当前在服务器中  将数据写到流中

FileOutputStream fos = new FileOutputStream("/Users/apple/Documents/复制品.JPG");

byte[] bytes = new byte[1024];

int len = 0 ;

while((len = in.read(bytes))!=-1){

fos.write(bytes, 0, len);

}

//5写完数据提示上传成功

s.getOutputStream().write("上传成功".getBytes());

//6释放资源

fos.close();

s.close();

server.close();

}

}

public class TCPCleint {

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

//1创建服务器对象

Socket s = new Socket("127.0.0.1", 9898);

//2指定路径

FileInputStream fis = new FileInputStream("/Users/apple/Desktop/1.JPG");

//3读取服务器数据

OutputStream out = s.getOutputStream();

byte[] bytes = new byte[1024];

int len = 0;

while((len = fis.read(bytes))!=-1) {

out.write(bytes,0,len);

}

//4告知书写完毕

s.shutdownOutput();

//5书写到服务器

InputStream in = s.getInputStream();

len = in.read(bytes);

System.out.println("服务器:"+new String(bytes,0,len));

//6释放资源

fis.close();

s.close();

}

}

## 练习三:上传图片文件

客户端需求：把一个图片文件发送到服务端并读取回馈信息。要求判断文件是否存在及格式是否为jpg并要求文件小于2M。

服务端需求：接收客户端发送过来的图片数据。进行存储后，回馈一个“上传成功”字样。支持多用户的并发访问。

客户端代码：

public class MyClient {  
  
 public static void main(String[] args) throws Exception {  
  
 File file = new File("D:\\aa.jpg");  
 if(!file.exists()){  
 return;  
 }  
 if (!file.getName().endsWith(".jpg")){  
 return;  
 }  
 if (file.length()>=1024\*1024\*2){  
  
 return;  
 }  
  
 Socket socket = new Socket("127.0.0.1",9999);  
  
 //输出流  
 OutputStream outputStream = socket.getOutputStream();  
 FileInputStream fileInputStream = new FileInputStream(file);  
 int len;  
 byte[] arr = new byte[1024];

while ((len=fileInputStream.read(arr))!=-1){  
 outputStream.write(arr,0,len);  
 }  
 socket.shutdownOutput();  
  
 InputStream inputStream = socket.getInputStream();  
 int i = inputStream.read(arr);  
 System.out.println(new String(arr,0,i));  
  
 fileInputStream.close();  
 outputStream.close();  
 socket.close();  
  
 }  
  
}

服务端代码：

public class MyServer {

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

ServerSocket serverSocket = new ServerSocket(9999);

while (true){

Socket socket = serverSocket.accept();

new Thread(){

@Override

public void run() {

try {

InputStream inputStream = socket.getInputStream();

FileOutputStream fileOutputStream = new FileOutputStream(System.currentTimeMillis()+".jpg");

int len;

byte[] arr = new byte[1024];

while ((len=inputStream.read(arr))!=-1){

fileOutputStream.write(arr,0,len);

}

OutputStream outputStream = socket.getOutputStream();

outputStream.write("上传成功".getBytes());

outputStream.close();

fileOutputStream.close();

inputStream.close();

socket.close();

}catch (Exception e){

e.printStackTrace();

}

}

}.start();

}

}

}