1. IMPLEMENTATION OF JAVA API ILLUSTRATING IPC USING TCP PROTOCOL.

TCPServer:

```
import java.io.*;
import java.net.*;
public class TCPServer
{
public static void main(String args[])
{
try
{
int serverport=7896;
ServerSocket listensocket=new ServerSocket(serverport);
while(true)
{
Socket clientsocket=listensocket.accept();
Connection c=new Connection(clientsocket);
}
}
catch(IOException e)
{
System.out.println("LISTEN:"+e.getMessage());
}
}
}
class Connection extends Thread
{
DataInputStream in;
DataOutputStream out;
Socket clientsocket;
public Connection(Socket aclientsocket)
{
try
{
clientsocket=aclientsocket;
in=new DataInputStream(clientsocket.getInputStream());
out=new DataOutputStream(clientsocket.getOutputStream());
this.start();
}
catch(IOException e)
{
System.out.println("Connection:"+e.getMessage());
}
}
public void run()
{
try
{
String data=in.readUTF();
out.writeUTF(data);
}
catch(EOFException e)
{
System.out.println("EOF:"+e.getMessage());
}
catch(IOException e)
{
System.out.println("IO:"+e.getMessage());
}
finally{
try
{
clientsocket.close();
}
catch(IOException e)
{
System.out.println("close failed");
}
}
}
}
```

TCPClient:

```
import java.net.*;
import java.io.*;
public class TCPClient
{
public static void main(String args[])
{
Socket s=null;
try
{
int serverport=7896;
s=new Socket(args[0],serverport);
DataInputStream in=new DataInputStream(s.getInputStream());
DataOutputStream out=new DataOutputStream(s.getOutputStream());
out.writeUTF(args[1]);
String data=in.readUTF();
System.out.println("Receive data from server:"+data);
}
catch(UnknownHostException e)
{
System.out.println("Socket:"+e.getMessage());
}
catch(EOFException e)
{
System.out.println("EOF:"+e.getMessage());
}
catch(IOException e)
{
System.out.println("IO:"+e.getMessage());
}
finally
{
if(s!=null)
try
{
s.close();
}
catch(IOException e)
{
System.out.println("close failed"); }}}}
```

2. IMPLEMENTATION OF JAVA API ILLUSTRATING IPC USING UDP PROTOCOL.

UDPServer:

```
import java.net.*;
import java.io.*;
public class UDPServer
{
public static void main(String args[])
{
DatagramSocket server=null;
try
{
server=new DatagramSocket(6789);
byte[] buffer=new byte[1000];
while(true)
{
DatagramPacket request=new DatagramPacket(buffer,buffer.length);
server.receive(request);
DatagramPacket reply=new
DatagramPacket(request.getData(),request.getLength(),request.getAddress(),reques
t.getPort());
server.send(reply);
}
}
catch(SocketException e)
{
System.out.println("Socket!"+e.getMessage());
}
catch(IOException e)
{System.out.println("IO:"+e.getMessage());
}
finally
{
if(server!=null)
server.close();
}
}
}
```

UDPClient:

```
import java.net.*;
import java.io.*;
public class UDPClient
{
public static void main(String args[])
{
DatagramSocket client=null;
try
{
client=new DatagramSocket();
byte[] m=args[1].getBytes();
InetAddress host=InetAddress.getByName(args[0]);
int serverport=6789;
DatagramPacket request=new DatagramPacket(m,args[1].length(),host,serverport);
client.send(request);
byte[] buffer=new byte[1000];
DatagramPacket reply=new DatagramPacket(buffer,buffer.length);
client.receive(reply);
String re=new String(reply.getData());
System.out.println("REPLY:"+re);
}
catch(SocketException e)
{
System.out.println("SoCKET!"+e.getMessage());
}
catch(IOException e)
{
System.out.println("IO:"+e.getMessage());
}
finally
{
if(client!=null)client.close();
}
}
}
```