TP4

Exercice 1 : Lire et interpréter des questions dans des datagrammes UDP

L'objectif de cet exercice est d'être capable d'écrire un programme capable de lire et d'interpréter des questions dans des datagrammes UDP. Un serveur de jeu est installé sur la machine de l'enseignant, le serveur a pour adresse IP 192.168.130.XX et écoute sur le port 7001 en UDP.

Avec netcat, vous pouvez lancer lancer des requêtes vers ce serveur.

Si vous envoyez le mot clé JOUER, le serveur vous répond ceci :

JOUER

Q10560:82+12=?

Code de base

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetSocketAddress;
/**
 * Client basique UDP
 */
public class ClientUDP
{
	public static void main(String[] args) throws Exception
	{
		ClientUDP clientUDP = new ClientUDP();
		clientUDP.execute();
	}
	/**
	 * Le client cree une socket, envoie un message UDP au serveur
	 */
	private void execute() throws IOException
	{
		//
		System.out.println("Demarrage du client ...");

		//Creation de la socket
		DatagramSocket socket = new DatagramSocket();

		// Creation et envoi du message à l'adresse 127.0.0.1 et le port 2000
		InetSocketAddress adrDest = new InetSocketAddress("192.168.130.24", 2000);
		byte[] bufE = new String("hello connard").getBytes();
		DatagramPacket dpE = new DatagramPacket(bufE, bufE.length, adrDest);
		socket.send(dpE);
		System.out.println("Message envoyé.");

		// Fermeture de la socket
		socket.close();
		System.out.println("Arret du client .");
	}
}
```

Lire et interpréter des questions dans des datagrammes UDP

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetSocketAddress;

public class ClientUDP {
	public static void main(String[] args) throws Exception
	{
		ClientUDP clientUDP = new ClientUDP();
		clientUDP.execute();
	}
	private void execute() throws IOException
	{
		String data[],equa[],idR,rep;
		int num1,num2;
		System.out.println("Demarrage du client ...");
		//Creation de la socket
		DatagramSocket socket = new DatagramSocket();
		// Creation et envoi du message à l'adresse 127.0.1 et le port 7001
		InetSocketAddress adrDest = new InetSocketAddress("127.0.1", 7001);
		byte[] bufE = new String("JOUER").getBytes();
		DatagramPacket dpE = new DatagramPacket(bufE, bufE.length, adrDest);
		socket.send(dpE);
		System.out.println("Message envoyé.");
		byte[] bufR = new byte[2048];
		DatagramPacket dpR = new DatagramPacket(bufR, bufR.length);
		socket.receive(dpR);
		String message = new String(bufR, dpR.getOffset(), dpR.getLength());
		data = message.split(":");
		idR  = "R" + data[0].substring(1,6);
		equa = data[1].split("\\+");
		num1 = Integer.parseInt(equa[0]);
		equa[1] = equa[1].substring(0,equa[1].length()-3);
		num2 = Integer.parseInt(equa[1]);
		rep = "R" + data[0].substring(1,6) + ":" + String.valueOf(num1+num2);
		System.out.println(rep);
		bufE = new String(rep).getBytes();
		dpE = new DatagramPacket(bufE, bufE.length, adrDest);
		socket.send(dpE);
		System.out.println("Message envoyé.");
		// Fermeture de la socket
		socket.close();
		System.out.println("Arret du client .");
	}

}
```

Lire et interpréter des questions dans un flux TCPimport java.io.IOException;

```
import java.io.InputStream;
import java.io.OutputStream;
import java.net.InetAddress;
import java.net.InetSocketAddress;
import java.net.Socket;

public class ClientTCP
{
	public static void main(String[] args) throws Exception
	{
		ClientTCP clientTCP = new ClientTCP();
		clientTCP.execute();
	}
	/**
	 * Le client cree une socket, envoie un message au serveur
	 * et attend la reponse
	 *
	 */
	private void execute() throws IOException
	{
		String message = "",temp ="", data[], equa[],rep = "";
		System.out.println("Demarrage du client ...");
		//Creation de la socket
		Socket socket = new Socket();
		// Connexion au serveur
		InetSocketAddress adrDest = new InetSocketAddress("127.0.0.1", 7500);
		socket.connect(adrDest);
		// Attente de la reponse
		byte[] bufR = new byte[2048];
		InputStream is = socket.getInputStream();
		while (!message.startsWith("close")) {
			int lenBufR = is.read(bufR);
			if (lenBufR!=-1)
			{
				message = new String(bufR, 0 , lenBufR );
				if (!message.endsWith("?")) {
					temp= new String(bufR, 0 , lenBufR );
					message = message + temp;
					System.out.println("message recue = "+temp);
				} else {
					System.out.println("message recue = "+message);
					data = message.split("\\?");
					data[0] = data[0].substring(0,data[0].length()-1);
					equa = data[0].split("\\+");
					rep = String.valueOf(Integer.parseInt(equa[0])
					 +Integer.parseInt(equa[1])) + ";";
					System.out.println("reponse = "+rep);
					byte[] bufE = new String(rep).getBytes();
					OutputStream os = socket.getOutputStream();

					os.write(bufE);
					System.out.println("Message envoye");
			}}}}}
```