**Programación sobre Redes - Guía de ejercicios 5**

**Compilador: Utilizar Eclipse/Netbeans/IntelliJ/VisualStudioCode**

**Ejercicio 1**

Desarrollar una aplicación cliente-servidor donde el cliente envíe un vector de enteros al servidor. El servidor debe calcular el valor máximo del vector y devolverlo al cliente, quien lo mostrará por pantalla.

Cliente:  
import java.io.IOException;

import java.io.PrintWriter;

import java.net.Socket;

import java.util.Scanner;

public class ClienteString {

public static void main(String[] args) {

try (Scanner in = new Scanner(System.***in***)) {

int vector[] = {1, 9, 17, 5, 2};

try (Socket socket = new Socket("localhost", 12346);

Scanner input = new Scanner(socket.getInputStream());

PrintWriter output = new PrintWriter(socket.getOutputStream())) {

// Envia el tamaño del vector al servidor

output.println(vector.length);

// Envia los valores del vector

for (int i = 0; i < vector.length; i++) {

output.println(vector[i]);

}

// Se asegura de que la información se envíe al servidor

output.flush();

//Recibe el valor máximo y la posición del servidor

int valorMaximo = input.nextInt();

int posMaximo = input.nextInt();

// Imprime el valor máximo y su posición

System.***out***.println("El máximo es " + valorMaximo + " y se encuentra en la posición " + posMaximo);

} catch (IOException e) {

e.printStackTrace();

}

}

}

}

echoClinete:

import java.io.IOException;

import java.io.PrintWriter;

import java.net.Socket;

import java.util.Scanner;

public class EchoClientHandlerr extends Thread {

private Socket socket;

public EchoClientHandlerr(Socket socket) {

this.socket = socket;

}

public void handle() {

this.start();

}

public void run() {

try (Scanner in = new Scanner(socket.getInputStream());

PrintWriter out = new PrintWriter(socket.getOutputStream(), true)) {

// Recibe el tamaño del vector

int tamanioVector = in.nextInt();

int[] vector = new int[tamanioVector];

//Recibe los valores del vector

for (int i = 0; i < tamanioVector; i++) {

vector[i] = in.nextInt();

}

// Calcula el valor máximo y su posición

int valorMaximo = vector[0];

int posMaximo = 0;

for (int i = 1; i < tamanioVector; i++) {

if (vector[i] > valorMaximo) {

valorMaximo = vector[i];

posMaximo = i;

}

}

//Envia el valor máximo y su posición al cliente

out.println(valorMaximo);

out.println(posMaximo);

out.flush();

} catch (Exception e) {

System.***err***.println("Error comunicándose con el cliente.");

e.printStackTrace();

} finally {

try {

if (this.socket != null) {

this.socket.close();

}

} catch (IOException e) {

System.***err***.println("Error cerrando el socket.");

}

}

}

}

**Ejercicio 2**

Desarrollar una aplicación cliente-servidor donde el cliente envíe un vector de números al servidor. El servidor debe calcular la suma de los elementos y devolverla al cliente, quien la mostrará por pantalla.

ClienteString.java

import java.io.IOException;

import java.io.PrintWriter;

import java.net.Socket;

import java.util.Scanner;

public class ClienteString {

public static void main(String[] args) {

try (Scanner in = new Scanner(System.in)) {

int vector[] = {1, 2, 3, 4, 5};

try (Socket socket = new Socket("localhost", 12345);

Scanner input = new Scanner(socket.getInputStream());

PrintWriter output = new PrintWriter(socket.getOutputStream())) {

output.println(vector.length);

for (int i = 0; i < vector.length; i++) {

output.println(vector[i]);

}

output.flush();

int suma = input.nextInt();

System.out.println("La suma de los elementos es: " + suma);

} catch (IOException e) {

e.printStackTrace();

}

}

}

}

EchoClientHandlerr.java

import java.io.IOException;

import java.io.PrintWriter;

import java.net.Socket;

import java.util.Scanner;

public class EchoClientHandlerr extends Thread {

private Socket socket;

public EchoClientHandlerr(Socket socket) {

this.socket = socket;

}

public void handle() {

this.start();

}

public void run() {

try (Scanner in = new Scanner(socket.getInputStream());

PrintWriter out = new PrintWriter(socket.getOutputStream(), true)) {

int tamanioVector = in.nextInt();

int[] vector = new int[tamanioVector];

for (int i = 0; i < tamanioVector; i++) {

vector[i] = in.nextInt();

}

int suma = 0;

for (int i = 0; i < tamanioVector; i++) {

suma += vector[i];

}

out.println(suma);

out.flush();

} catch (Exception e) {

System.err.println("Error comunicándose con el cliente.");

e.printStackTrace();

} finally {

try {

if (this.socket != null) {

this.socket.close();

}

} catch (IOException e) {

System.err.println("Error cerrando el socket.");

}

}

}

}

**Ejercicio 3**

Desarrollar una aplicación cliente-servidor donde el cliente envíe una cadena de texto al servidor. El servidor debe contar cuántas vocales tiene el texto y devolver el resultado al cliente, quien lo mostrará por pantalla.

ClienteString.java  
import java.io.IOException;

import java.io.PrintWriter;

import java.net.Socket;

import java.util.Scanner;

public class ClienteString {

public static void main(String[] args) {

try (Scanner in = new Scanner(System.in)) {

System.out.print("Ingrese una cadena de texto: ");

String texto = in.nextLine();

try (Socket socket = new Socket("localhost", 12345);

Scanner input = new Scanner(socket.getInputStream());

PrintWriter output = new PrintWriter(socket.getOutputStream())) {

output.println(texto);

output.flush();

int cantidadVocales = input.nextInt();

System.out.println("La cantidad de vocales en la cadena es: " + cantidadVocales);

} catch (IOException e) {

e.printStackTrace();

}

}

}

**EchoClientHandlerr.java**

**import java.io.IOException;**

**import java.io.PrintWriter;**

**import java.net.Socket;**

**import java.util.Scanner;**

**public class EchoClientHandlerr extends Thread {**

**private Socket socket;**

**public EchoClientHandlerr(Socket socket) {**

**this.socket = socket;**

**}**

**public void handle() {**

**this.start();**

**}**

**public void run() {**

**try (Scanner in = new Scanner(socket.getInputStream());**

**PrintWriter out = new PrintWriter(socket.getOutputStream(), true)) {**

**String texto = in.nextLine();**

**int cantidadVocales = contarVocales(texto);**

**out.println(cantidadVocales);**

**out.flush();**

**} catch (Exception e) {**

**System.err.println("Error comunicándose con el cliente.");**

**e.printStackTrace();**

**} finally {**

**try {**

**if (this.socket != null) {**

**this.socket.close();**

**}**

**} catch (IOException e) {**

**System.err.println("Error cerrando el socket.");**

**}**

**}**

**}**

**private int contarVocales(String texto) {**

**int contador = 0;**

**String vocales = "aeiouAEIOU";**

**for (int i = 0; i < texto.length(); i++) {**

**if (vocales.indexOf(texto.charAt(i)) != -1) {**

**contador++;**

**}**

**}**

**return contador;**

**}**

**}**

**Ejercicio 4**

Desarrollar una aplicación cliente-servidor donde el cliente envíe dos vectores de la misma longitud al servidor. El servidor debe calcular el producto escalar de ambos vectores (sumatoria de las multiplicaciones de sus elementos) y devolver el resultado al cliente, quien lo mostrará por pantalla.

ClienteString.java

import java.io.IOException;

import java.io.PrintWriter;

import java.net.Socket;

import java.util.Scanner;

public class ClienteString {

public static void main(String[] args) {

int[] vector1 = {1, 2, 3}; // Primer vector

int[] vector2 = {4, 5, 6}; // Segundo vector

try (Socket socket = new Socket("localhost", 12345);

Scanner input = new Scanner(socket.getInputStream());

PrintWriter output = new PrintWriter(socket.getOutputStream())) {

output.println(vector1.length);

output.flush();

for (int num : vector1) {

output.println(num);

}

for (int num : vector2) {

output.println(num);

}

output.flush();

int productoEscalar = input.nextInt();

System.out.println("El producto escalar es: " + productoEscalar);

} catch (IOException e) {

e.printStackTrace();

}

}

}

EchoClientHandlerr.java

import java.io.IOException;

import java.io.PrintWriter;

import java.net.Socket;

import java.util.Scanner;

public class EchoClientHandlerr extends Thread {

private Socket socket;

public EchoClientHandlerr(Socket socket) {

this.socket = socket;

}

public void handle() {

this.start();

}

public void run() {

try (Scanner in = new Scanner(socket.getInputStream());

PrintWriter out = new PrintWriter(socket.getOutputStream(), true)) {

int tamanioVector = in.nextInt();

int[] vector1 = new int[tamanioVector];

int[] vector2 = new int[tamanioVector];

for (int i = 0; i < tamanioVector; i++) {

vector1[i] = in.nextInt();

}

for (int i = 0; i < tamanioVector; i++) {

vector2[i] = in.nextInt();

}

int productoEscalar = 0;

for (int i = 0; i < tamanioVector; i++) {

productoEscalar += vector1[i] \* vector2[i];

}

out.println(productoEscalar);

out.flush();

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

if (this.socket != null) {

this.socket.close();

}

} catch (IOException e) {

e.printStackTrace();

}

}

}

}

**Ejercicio 5**

Desarrollar una aplicación cliente-servidor donde el cliente envíe un array de enteros al servidor. El servidor debe invertir el array y devolverlo al cliente, quien lo mostrará por pantalla.

ClienteString.java

import java.io.IOException;

import java.io.PrintWriter;

import java.net.Socket;

import java.util.Scanner;

public class ClienteString {

public static void main(String[] args) {

int[] vector = {1, 2, 3, 4, 5}; // Vector a invertir

try (Socket socket = new Socket("localhost", 12345);

Scanner input = new Scanner(socket.getInputStream());

PrintWriter output = new PrintWriter(socket.getOutputStream())) {

output.println(vector.length);

output.flush();

for (int num : vector) {

output.println(num);

}

output.flush();

int[] vectorInvertido = new int[vector.length];

for (int i = 0; i < vector.length; i++) {

vectorInvertido[i] = input.nextInt();

}

System.out.print("El vector invertido es: ");

for (int num : vectorInvertido) {

System.out.print(num + " ");

}

} catch (IOException e) {

e.printStackTrace();

}

}

}

EchoClientHandlerr.java

import java.io.IOException;

import java.io.PrintWriter;

import java.net.Socket;

import java.util.Scanner;

public class EchoClientHandlerr extends Thread {

private Socket socket;

public EchoClientHandlerr(Socket socket) {

this.socket = socket;

}

public void handle() {

this.start();

}

public void run() {

try (Scanner in = new Scanner(socket.getInputStream());

PrintWriter out = new PrintWriter(socket.getOutputStream(), true)) {

int tamanioVector = in.nextInt();

int[] vector = new int[tamanioVector];

for (int i = 0; i < tamanioVector; i++) {

vector[i] = in.nextInt();

}

int[] vectorInvertido = new int[tamanioVector];

for (int i = 0; i < tamanioVector; i++) {

vectorInvertido[i] = vector[tamanioVector - i - 1];

}

for (int num : vectorInvertido) {

out.println(num);

}

out.flush();

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

if (this.socket != null) {

this.socket.close();

}

} catch (IOException e) {

e.printStackTrace();

}

}

}

}

**Ejercicio 6**

Desarrollar una aplicación cliente-servidor donde el cliente envíe un vector de enteros o decimales al servidor. El servidor debe calcular el promedio de los elementos y devolverlo al cliente, quien lo mostrará por pantalla.

ClienteString.java

import java.io.IOException;

import java.io.PrintWriter;

import java.net.Socket;

import java.util.Scanner;

public class ClienteString {

public static void main(String[] args) {

double[] vector = {1.5, 2.0, 3.5, 4.0, 5.5}; // Vector de números

try (Socket socket = new Socket("localhost", 12345);

Scanner input = new Scanner(socket.getInputStream());

PrintWriter output = new PrintWriter(socket.getOutputStream())) {

output.println(vector.length);

output.flush();

for (double num : vector) {

output.println(num);

}

output.flush();

double promedio = input.nextDouble();

System.out.println("El promedio de los elementos es: " + promedio);

} catch (IOException e) {

e.printStackTrace();

}

}

}

EchoClientHandlerr.java

import java.io.IOException;

import java.io.PrintWriter;

import java.net.Socket;

import java.util.Scanner;

public class EchoClientHandlerr extends Thread {

private Socket socket;

public EchoClientHandlerr(Socket socket) {

this.socket = socket;

}

public void handle() {

this.start();

}

public void run() {

try (Scanner in = new Scanner(socket.getInputStream());

PrintWriter out = new PrintWriter(socket.getOutputStream(), true)) {

int tamanioVector = in.nextInt();

double[] vector = new double[tamanioVector];

for (int i = 0; i < tamanioVector; i++) {

vector[i] = in.nextDouble();

}

double suma = 0;

for (double num : vector) {

suma += num;

}

double promedio = suma / tamanioVector;

out.println(promedio);

out.flush();

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

if (this.socket != null) {

this.socket.close();

}

} catch (IOException e) {

e.printStackTrace();

}

}

}

}

**Ejercicio 7**

Desarrollar una aplicación cliente-servidor donde el cliente envíe un vector desordenado al servidor. El servidor debe ordenar el vector en orden ascendente y devolverlo al cliente, quien lo mostrará por pantalla.

ClienteString.java

import java.io.IOException;

import java.io.PrintWriter;

import java.net.Socket;

import java.util.Scanner;

public class ClienteString {

public static void main(String[] args) {

int[] vector = {5, 3, 8, 1, 2}; // Vector desordenado

try (Socket socket = new Socket("localhost", 12345);

Scanner input = new Scanner(socket.getInputStream());

PrintWriter output = new PrintWriter(socket.getOutputStream())) {

output.println(vector.length);

output.flush();

for (int num : vector) {

output.println(num);

}

output.flush();

int[] vectorOrdenado = new int[vector.length];

for (int i = 0; i < vector.length; i++) {

vectorOrdenado[i] = input.nextInt();

}

System.out.print("El vector ordenado es: ");

for (int num : vectorOrdenado) {

System.out.print(num + " ");

}

} catch (IOException e) {

e.printStackTrace();

}

}

}

EchoClientHandlerr.java

import java.io.IOException;

import java.io.PrintWriter;

import java.net.Socket;

import java.util.Arrays;

import java.util.Scanner;

public class EchoClientHandlerr extends Thread {

private Socket socket;

public EchoClientHandlerr(Socket socket) {

this.socket = socket;

}

public void handle() {

this.start();

}

public void run() {

try (Scanner in = new Scanner(socket.getInputStream());

PrintWriter out = new PrintWriter(socket.getOutputStream(), true)) {

int tamanioVector = in.nextInt();

int[] vector = new int[tamanioVector];

for (int i = 0; i < tamanioVector; i++) {

vector[i] = in.nextInt();

}

Arrays.sort(vector);

for (int num : vector) {

out.println(num);

}

out.flush();

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

if (this.socket != null) {

this.socket.close();

}

} catch (IOException e) {

e.printStackTrace();

}

}

}

}

**Ejercicio 8**

Desarrollar una aplicación cliente-servidor donde el cliente envíe una cadena de texto al servidor. El servidor debe contar cuántas palabras tiene el texto y devolver el resultado al cliente, quien lo mostrará por pantalla.

ClienteString.java

import java.io.IOException;

import java.io.PrintWriter;

import java.net.Socket;

import java.util.Scanner;

public class ClienteString {

public static void main(String[] args) {

try (Scanner in = new Scanner(System.in)) {

// Pedir al usuario una cadena de texto

System.out.print("Ingrese una cadena de texto: ");

String texto = in.nextLine();

try (Socket socket = new Socket("localhost", 12345);

Scanner input = new Scanner(socket.getInputStream());

PrintWriter output = new PrintWriter(socket.getOutputStream())) {

output.println(texto);

output.flush();

int numeroPalabras = input.nextInt();

System.out.println("La cantidad de palabras en la cadena es: " + numeroPalabras);

} catch (IOException e) {

e.printStackTrace();

}

}

}

}

EchoClientHandlerr.java

import java.io.IOException;

import java.io.PrintWriter;

import java.net.Socket;

import java.util.Scanner;

public class EchoClientHandlerr extends Thread {

private Socket socket;

public EchoClientHandlerr(Socket socket) {

this.socket = socket;

}

public void handle() {

this.start();

}

public void run() {

try (Scanner in = new Scanner(socket.getInputStream());

PrintWriter out = new PrintWriter(socket.getOutputStream(), true)) {

String texto = in.nextLine();

int numeroPalabras = texto.trim().isEmpty() ? 0 : texto.trim().split("\\s+").length;

out.println(numeroPalabras);

out.flush();

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

if (this.socket != null) {

this.socket.close();

}

} catch (IOException e) {

e.printStackTrace();

}

}

}

}

**Ejercicio 9**

Desarrollar una aplicación cliente-servidor donde el cliente envíe una cantidad de dinero y una moneda de origen al servidor. El servidor debe convertir esa cantidad a una moneda destino predefinida (por ejemplo, de pesos a dólares) y devolver el resultado al cliente, quien lo mostrará por pantalla.

ClienteString.java

import java.io.IOException;

import java.io.PrintWriter;

import java.net.Socket;

import java.util.Scanner;

public class ClienteString {

public static void main(String[] args) {

try (Scanner in = new Scanner(System.in)) {

// Pedir cantidad y moneda de origen

System.out.print("Ingrese la cantidad de dinero: ");

double cantidad = in.nextDouble();

in.nextLine(); // Limpiar el buffer

System.out.print("Ingrese la moneda de origen (USD): ");

String monedaOrigen = in.nextLine();

try (Socket socket = new Socket("localhost", 12345);

Scanner input = new Scanner(socket.getInputStream());

PrintWriter output = new PrintWriter(socket.getOutputStream())) {

output.println(cantidad);

output.println(monedaOrigen);

output.flush();

double valorConvertido = input.nextDouble();

System.out.println("El valor convertido es: " + valorConvertido);

} catch (IOException e) {

e.printStackTrace();

}

}

}

}

EchoClientHandlerr.java

import java.io.IOException;

import java.io.PrintWriter;

import java.net.Socket;

import java.util.Scanner;

public class EchoClientHandlerr extends Thread {

private Socket socket;

public EchoClientHandlerr(Socket socket) {

this.socket = socket;

}

public void handle() {

this.start();

}

public void run() {

try (Scanner in = new Scanner(socket.getInputStream());

PrintWriter out = new PrintWriter(socket.getOutputStream(), true)) {

double cantidad = in.nextDouble();

String monedaOrigen = in.nextLine().trim();

double valorConvertido = 0;

if (monedaOrigen.equalsIgnoreCase("USD")) {

valorConvertido = cantidad \* 0.85;

}

out.println(valorConvertido);

out.flush();

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

if (this.socket != null) {

this.socket.close();

}

} catch (IOException e) {

e.printStackTrace();

}

}

}

}

**Ejercicio 10**

Desarrollar una aplicación cliente-servidor donde el cliente envíe una matriz al servidor. El servidor debe primero calcular la matriz transpuesta y luego multiplicar la matriz original por su transpuesta. Finalmente, el servidor devolverá la matriz resultante de esta multiplicación al cliente, quien la mostrará por pantalla. El cliente debe asegurarse de que la matriz sea adecuada para la operación antes de enviarla (por ejemplo, que sea de dimensiones adecuadas para la multiplicación).

ClienteString.java

import java.io.IOException;

import java.io.PrintWriter;

import java.net.Socket;

import java.util.Scanner;

public class ClienteString {

public static void main(String[] args) {

int[][] matriz = {

{1, 2},

{3, 4}

};

try (Socket socket = new Socket("localhost", 12345);

Scanner input = new Scanner(socket.getInputStream());

PrintWriter output = new PrintWriter(socket.getOutputStream())) {

output.println(matriz.length);

output.println(matriz[0].length);

output.flush();

for (int i = 0; i < matriz.length; i++) {

for (int j = 0; j < matriz[0].length; j++) {

output.println(matriz[i][j]);

}

}

output.flush();

System.out.println("Matriz resultante:");

for (int i = 0; i < matriz.length; i++) {

for (int j = 0; j < matriz.length; j++) {

System.out.print(input.nextInt() + " ");

}

System.out.println();

}

} catch (IOException e) {

e.printStackTrace();

}

}

}

EchoClientHandlerr.java

import java.io.IOException;

import java.io.PrintWriter;

import java.net.Socket;

import java.util.Scanner;

public class EchoClientHandlerr extends Thread {

private Socket socket;

public EchoClientHandlerr(Socket socket) {

this.socket = socket;

}

public void handle() {

this.start();

}

public void run() {

try (Scanner in = new Scanner(socket.getInputStream());

PrintWriter out = new PrintWriter(socket.getOutputStream(), true)) {

int filas = in.nextInt();

int columnas = in.nextInt();

int[][] matriz = new int[filas][columnas];

for (int i = 0; i < filas; i++) {

for (int j = 0; j < columnas; j++) {

matriz[i][j] = in.nextInt();

}

}

int[][] transpuesta = new int[columnas][filas];

for (int i = 0; i < filas; i++) {

for (int j = 0; j < columnas; j++) {

transpuesta[j][i] = matriz[i][j];

}

}

int[][] resultado = new int[filas][filas];

for (int i = 0; i < filas; i++) {

for (int j = 0; j < filas; j++) {

for (int k = 0; k < columnas; k++) {

resultado[i][j] += matriz[i][k] \* transpuesta[k][j];

}

}

}

for (int i = 0; i < filas; i++) {

for (int j = 0; j < filas; j++) {

out.println(resultado[i][j]);

}

}

out.flush();

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

if (this.socket != null) {

this.socket.close();

}

} catch (IOException e) {

e.printStackTrace();

}

}

}

}