Programación de Sistemas Distribuidos Práctica 1: Comunicación con Sockets Arturo Requejo Portilla





Tabla de contenido

I. Enunciados		3
A. Ejercicio 1	3	
B. Ejercicio 2		
II. Resultados		<i>3</i>
a. Introducción	3	
A. Ejercicio 1	3	
B. Ejercicio 2	4	
IV. Conclusiones		5



Enunciados

A. Ejercicio 1

Implementar un programa que sea capaz de calcular el área de un círculo. Para ello, será necesario crear la clase Circulo e implementar el método dentro de esta clase. Los parámetros se pasarán por referencia. **Justificar el proceso 4 PUNTOS.**

B. Ejercicio 2

Implementar la clase Circulo desarrollada en el ejercicio 1 en un servidor y realizar la petición de cálculo del área desde el cliente. La comunicación se realizará a través de la implementación de sockets. **Justificar el proceso 6 PUNTOS.**

II. Resultados

a) Introducción

Para comprender esta práctica primero debemos entender lo que es un socket y su funcionamiento. Un socket es un mecanismo que permite intercambiar a 2 procesos cualquier flujo de datos permitiendo así que se comuniquen entre ellos.

Una vez aprendido lo que es un socket de forma teórica ahora entendamos el código.

En la clase servidor empezamos creando un servidor socket, permitiendo establecer una comunicación con el cliente, donde al principio lo ponemos a null pero luego establece un puerto donde ambos se comunicaran. La clase socket permite asignar un punto de llegada de los datos para así poder establecer la comunicación y por último el datainputstream y data outputstream permite tanto leer como escribir respectivamente y para finalizar iniciamos el puerto y creamos un while haciendo que el servidor este siempre en escucha.

En el lado del cliente empezamos definiendo tanto el puerto (que tiene que coincidir con el servidor si no no hay comunicación) y host (siendo la ip del cliente). Con el in y out permitimos la entrada y salida del socket (mismo funcionamiento en el servidor).

A. Ejercicio 1

En este ejercicio debemos realizar una clase circulo donde realizamos una función que calcule el área de un círculo a partir de un radio. Para ello empezamos creando un constructor donde cuando inicialicemos un objeto circulo tengamos que indicar el radio con el que vamos a calcular el área. Luego por último en esta clase vamos a crear una función área donde pasamos como parámetro de entrada el radio con el que calculemos el área. La fórmula para calcular el área de un círculo es:

 $Área = PI * (radio)^2$

Para ello en la función área empezamos inicializando la variable double área donde almacenaremos el resultado del cálculo, luego mediante la librería Math sacamos el numero PI exacto haciendo el resultado lo más preciso posible y por último lo multiplicamos por el radio introducido al cuadrado.

```
public class Circulo {
    public double radio;

    public Circulo(double radio) {
        radio = this.radio;
    }

    public double area(double radio) {
        double area;
        area = Math.PI * Math.pow(radio, 2);
        return area;
    }
}
```

B. Ejercicio 2

Para este segundo ejercicio debemos hacer que el servidor calcule el área que recibe por parte del cliente.

Para ello empezamos enviando el radio a través del cliente. En el cliente imprimimos el radio con el que vamos a calcular el área, posteriormente a través de la línea out.writeDouble(radio) enviaremos el radio al servidor.

En el servidor recibimos el radio enviado por el cliente a través de la línea in.readDouble donde almacenaremos el radio en la variable double mensaje. El servidor imprime el radio recibido para así saber si hemos recibido el mensaje correctamente. Luego creamos un objeto circulo inicializando con el radio recibido. Luego calculamos el radio atreves de la función área(mensaje) donde calcularemos el área y la almacenaremos en la variable double área. Imprimimos el resultado en el servidor y por último mediante la función writeDouble(area) enviaremos el área calculada al cliente.

En el cliente leemos el mensaje enviado por el servidor, siendo este mensaje el área calculada a partir del radio enviado por el cliente, una vez recibido lo almacenamos en la variable mensaje donde el cliente imprime por pantalla el área calculada por el servidor.

```
Servidor [Java Application] /snap/eclipse/66/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_17.0.5.v20221102-0933/jre/bin/java (Feb 6, 2023, 6:19:21 PM) [pid: 4657] Servidor iniciado
```

Como podemos ver la imagen superior podemos ver como inicializamos el servidor y está a la escucha activa de peticiones por parte de cliente.

```
<terminated> Cliente [Java Application] /snap/eclipse/66/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_17.0.5.v20221102-0933/jre/bin/java (Feb 6, 2023, 6:19:39 PM – 6:19:39 PM) [pid: 4713 EL radio es: 4.0
```

En esta imagen vemos como ya el cliente ha enviado la petición de cálculo de área con un radio determinado, lo imprimimos en la pantalla del cliente para saber si el valor del radio es correcto.

```
Servidor [Java Application]/snap/eclipse/66/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_17.0.5.v20221102-0933/jre/bin/java (Feb 6, 2023, 6:19:21 PM) [pid: 4657]
Servidor iniciado
Cliente conectado
El radio con el que calcularemos el area es: 4.0
El resultado mandado es: 50.26548245743669
Cliente desconectado
```

En esta penúltima imagen vemos como el servidor ha recibido correctamente la petición del cliente y en el mismo servidor calculamos el área del círculo en función del dato introducido por el cliente y así mismo enviamos el resultado al cliente.

```
-terminated> Cliente [Java Application] /snap/eclipse/66/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_17.0.5.v20221102-0933/jre/bin/java (Feb 6, 2023, 6:19:39 PM – 6:19:39 PM) [pid: 4713] El radio es: 4.0
El area es: 50.26548245743669
```



En esta última imagen vemos como el cliente ha recibido correctamente el resultado del cálculo y lo imprimimos por pantalla. No es del todo correcto imprimir constantemente lo que envías y recibes tanto en cliente como servidor, pero quería demostrar que la comunicación se realiza de forma correcta.

C. Conclusiones

En esta práctica hemos aprendido lo que es un socket y su funcionamiento, a intercambiar mensajes entres un cliente y un servidor de manera local. No mandamos el objeto completo circulo al cliente debido a que como hemos visto en la teoría si mandásemos todo el objeto seria muy poco seguro. La descompresión del objeto se haría en el proxy chequeando previamente si el cliente ha pasado la autenticación, al no tener proxy pasaremos solo el resultado del calculo del área.

Resumidamente y en línea generales el cliente manda al servidor una petición de cálculo de área atreves del radio, el servidor recibe este dato y llamando al objeto círculo calcula el área en base a ese radio, una vez calculado el servidor devuelve SOLO el área y no todo el objeto.

D. Bibliografía

- -https://docs.oracle.com/javase/7/docs/api/java/net/ServerSocket.html
- -https://www.javatpoint.com/socket-programming