

Java – S08

Alin Zamfiroiu

S08 – Client–server

- ▶ Se construiește un proiect java cu doua pachete: **client** si **server**.
- ▶ In pachetul **client** se creeaza o clasa cu functia statica void main();
- ▶ In cadrul functiei se creeaza un obiect de tipul **Socket** (java.net.Socket)
- ▶ Acesta primeste ca si parametrii numele serverului si numarul portului prin care comunica cu serverul.

S08 – Client–server

```
public class ClientMain {  
    public static void main(String[] args) {  
        try {  
            Socket client =new Socket("localhost", 5000);
```

- ▶ Obiectul de tip socket are metoda `getOutputStream()`, care returneaza un obiect de tipul `OutputStream`.
- ▶ Se defineste un obiect `DataOutputStream`. Constructorul primeste obiectul de tip `OutputStream`. Acesta are metode de scriere care vor trimite fluxuri de date catre server.

S08 - Client-server

```
OutputStream os=client.getOutputStream();  
DataOutputStream dos=new DataOutputStream(os);  
dos.writeUTF("Salutari de la client");
```

S08 – Client–server

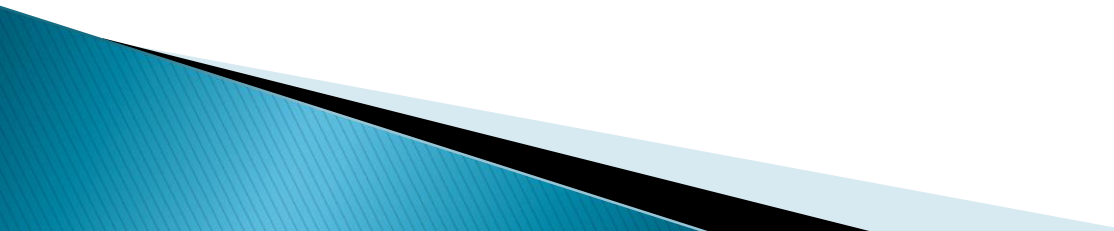
- ▶ In pachetul **server** se creeaza o clasa cu functia statica void main() si care implementeaza **Runnable**; Aceasta are un atribut static de tip **ServerSocket**
- ▶ In cadrul functiei main se initializeaza atributul static de tip **ServerSocket**.
Constructorul primeste ca parametru numarul portului ce urmeaza a fi deschis pentru ascultare.

S08 – Client–server

```
public class ServerMain implements Runnable {  
  
    private static ServerSocket serverSocket;  
  
    public static void main(String[] args) {  
  
        try {  
            serverSocket=new ServerSocket(5000);  
        } catch (IOException e) {  
            // TODO Auto-generated catch block  
            e.printStackTrace();  
        }  
    }  
}
```

- ▶ Tot in main se creeaza un Thread pe baza unui obiect de tip ServerMain si se porneste acest fir de executie.

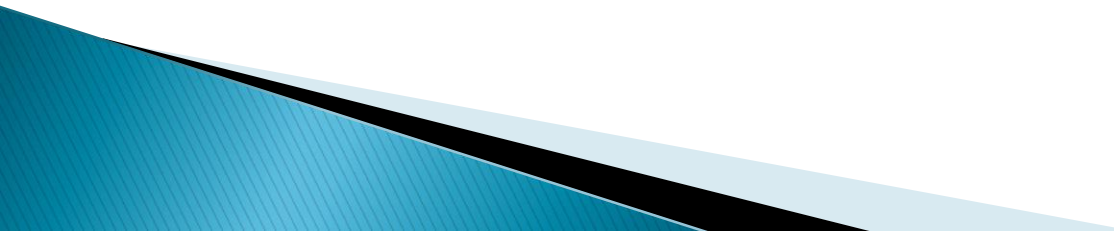
S08 – Client–server

- ▶ In metoda run() se creeaza un obiect de tip Socket care este initializat prin metoda accept a obiectului de tip ServerSocket.
 - ▶ Se creeaza un obiect de tip InputStream si unul de tip DataInputStream si se citeste prin intermediul metodelor de citire specifice obiectelor DataInputStream.
 - ▶ Acest lucru se face la infinit.
- 

S08 - Client-server

```
public void run() {
    while(true){
        try {
            Socket server=serverSocket.accept();
            InputStream is=server.getInputStream();
            DataInputStream dis=new DataInputStream(is);
            System.out.println("Server side: "+dis.readUTF());
            server.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```


S08 – Client–server

- ▶ Sa se implementeze pe partea de client trimiterea de informatii catre server dar si primirea informatiilor de la server. La fel pe partea de server sa se implementeze primirea de informatii respectiv trimiterea de informatii catre client.
- 

S08 – Client–server

- ▶ Sa se realizeze conectarea mai multor clienti la acelasi server si sa trimita mesaje. Serverul salveaza arhiva mesajelor si la primirea unei comenzi de genul “get” de la client sa trimita toata arhiva de mesaje.

