**Client-Server Product Purchase System**

**Description**

This Java application implements a client-server system where multiple clients interact with a central server to purchase products based on dynamically generated prices.

* The **server** periodically generates a random product price (between €10 and €100) and sends it to each client.
* Each **client** randomly generates a maximum price it is willing to pay (between €10 and €75) and decides whether to purchase or not.
* Each client makes up to 10 purchases. After completing all purchases, the client notifies the server and shuts down.
* The server keeps accepting clients and handles them concurrently using a thread pool. It logs all client interactions.

All source code, comments, and documentation are written in English, as required.

**File Structure**

* **Server.java**: Launches the server, accepts connections, and assigns each to a ClientHandler.
* **ClientHandler.java**: Handles communication with a single client in its own thread.
* **Client.java**: Connects to the server, receives prices, makes purchase decisions, and terminates after 10 purchases.
* **LauncherClients.java**: Prompts the user to specify how many clients to start, and launches them in parallel threads.
* **Request.java**: Serializable message sent from client to server, carrying purchase decisions.
* **Response.java**: Serializable message sent from server to client, containing current item price.

**How to Compile**

Ensure all .java files are in the same directory.

*javac \*.java*

**How to Run the Application**

1. **Start the server**

In one terminal, execute:

*java Server*

You should see:

Server ready, listening on port 8354

1. **Launch the clients**

In a **separate terminal**, run:

*java LauncherClients*

You will be prompted:

*How many clients do you want to launch? [max 10]:*

Enter a number (e.g., 3), and you'll see each client start, connect, and interact with the server.

Each client will:

* Receive a product price.
* Generate a maximum price it is willing to pay.
* Decide whether to purchase.
* Repeat until it buys 10 items.

Once done, each client shuts down gracefully.

**Example Output**

Client 1: Connected to server.

Client 1: Product price: 45€, Max price: 60€

Client 1: Purchased! Total: 1

...

Client 1: Finished purchasing. Shutting down.

**Notes**

* Clients and server communicate using serialized Java objects (Request, Response).
* Each ClientHandler runs in a separate thread from a fixed thread pool.
* Input validation is included when asking for the number of clients.
* Comments and Javadoc are written in English throughout the codebase.

**Requirements**

* Java 11 or higher (for modern syntax and features)