**Client-Server Communication Using Sockets**

This overview explains how to implement client-server communication using sockets in Java. Sockets provide a way for two programs to communicate over a network, allowing them to exchange data.

**Algorithm Overview**

* Create a server that listens for incoming connections on a specific port.
* Create a client that connects to the server using the server's IP address and port.
* Establish a connection between the server and client using sockets.
* Exchange data between the server and client using input and output streams.
* Close the connection and sockets when the communication is complete.

**Server-Side Implementation**

* Create a ServerSocket that listens for incoming connections on a specific port.
* Wait for a client to connect using the accept() method, which returns a Socket() for the established connection.
* Obtain input and output streams from the socket to read and write data.
* Process the received data and send a response to the client using the output stream.
* Close the socket and input/output streams when the communication is finished.

**Client-Side Implementation**

* Create a Socket to connect to the server using the server's IP address and port.
* Obtain input and output streams from the socket to read and write data.
* Send data to the server using the output stream.
* Read the server's response using the input stream.
* Close the socket and input/output streams when the communication is complete.