ㅣ

최정원

# github URL (optional):

|  |  |
| --- | --- |
|  | [chlwjddnjs/Calculator-in-Cloud-with-Java: calculator that cooperate with server and client with java (github.com)](https://github.com/chlwjddnjs/Calculator-in-Cloud-with-Java) |

# #개요

|  |  |
| --- | --- |
|  | Calculator in the Cloud  With Java  . Client Side (클라이언트 측):  • 사용자는 클라이언트에게 수식을 입력합니다.  • 클라이언트는 입력된 수식을 서버로 전송합니다.  • 사용자가 "bye"를 입력하여 연결을 종료할 때까지 계속 수식을 입력하고 결과를 받습니다.  . Server Side (서버 측):  • 서버는 클라이언트의 연결을 기다립니다.  • 클라이언트로부터 수식을 받아 계산합니다.  • 계산 결과를 다시 클라이언트에게 전송합니다.  • 클라이언트가 "bye"를 보내 연결을 종료할 때까지 계속 수식을 받고 결과를 전송합니다.  . Communication Flow (통신 흐름):  • 클라이언트가 서버로 수식을 전송합니다.  • 서버는 수식을 받아 계산하고 그 결과를 다시 클라이언트로 보냅니다.  • 이러한 요청과 응답 과정이 반복됩니다.  • 클라이언트가 "bye"를 입력하여 연결을 종료할 수 있습니다. |

# PRotocol

|  |  |
| --- | --- |
|  | 클라이언트-서버 프로토콜:  . 클라이언트 동작:  • 클라이언트는 사용자로부터 "operand operator operand" 형식의 수식을 입력받습니다.  • 입력된 수식은 서버로 전송됩니다.  • "bye"를 입력하면 클라이언트는 종료됩니다.  . 서버 동작:  • 서버는 클라이언트의 연결을 수락하고 클라이언트로부터 수식을 받습니다.  • 받은 수식을 처리하여 계산한 결과를 다시 클라이언트에게 전송합니다.  • "bye"를 받으면 클라이언트와의 연결을 종료합니다.  . 통신 Flow:  • 클라이언트가 수식을 서버로 전송합니다.  • 서버는 수식을 받아 계산하고 결과를 다시 클라이언트로 전송합니다.  • 이 과정을 클라이언트가 "bye"를 입력하여 연결을 종료할 때까지 반복합니다. |

## SOURCE CODES(client)

|  |
| --- |
| **package**myPackage;  **import**java.io.DataInputStream;  **import**java.io.DataOutputStream;  **import**java.io.IOException;  **import**java.net.InetAddress;  **import**java.net.Socket;  **import** java.net.UnknownHostException;  **import**java.util.Scanner;  **public** **class**calculator\_client {  **public** **static** **void**main(String[] args) **throws**IOException {  // Get the IP address of the local host  InetAddress ip= InetAddress.*getLocalHost*();  // Define the port number to establish the connection  **int** port= 4444;  // Create a Scanner object to read user input  Scanner sc= **new**Scanner(System.***in***);  // Create a socket to connect to the server  Socket s= **new**Socket(ip, port);  // Create input and output streams for communication with the server  DataInputStream dis= **new**DataInputStream(s.getInputStream());  DataOutputStream dos= **new**DataOutputStream(s.getOutputStream());  // Continuous loop to interact with the server  **while**(**true**) {  // Prompt the user to enter an equation in a specific format  System.***out***.print("Enter the equation in the form: ");  System.***out***.println("'operand operator operand'");  // Read the user input  String inp= sc.nextLine();  // Check if the user wants to exit  **if**(inp.equals("bye"))  **break**;  // Send the input equation to the server  dos.writeUTF(inp);  // Receive and display the result from the server  String ans= dis.readUTF();  System.***out***.println("Answer="+ ans);  }  }  } |

SOURCE CODES(server)

|  |
| --- |
| **package**myPackage;  **import**java.io.DataInputStream;  **import**java.io.DataOutputStream;  **import**java.io.IOException;  **import**java.net.ServerSocket;  **import**java.net.Socket;  **import**java.util.StringTokenizer;  **public** **class**calculator\_server {  **public** **static** **void**main(String args[]) **throws**IOException {  // Create a server socket with port 4444  ServerSocket ss= **new**ServerSocket(4444);  // Accept incoming client connection  Socket s= ss.accept();  // Create input and output streams for communication with the client  DataInputStream dis= **new**DataInputStream(s.getInputStream());  DataOutputStream dos= **new**DataOutputStream(s.getOutputStream());  // Continuous loop to handle incoming requests  **while**(**true**) {  // Read the input from the client  String input= dis.readUTF();  // Check if the client wants to exit  **if**(input.equals("bye"))  **break**;  // Display the received equation  System.***out***.println("Equation received: "+ input);  **int** result;  // Tokenizethe input to get operands and operation  StringTokenizer st= **new**StringTokenizer(input);  **int** oprnd1= Integer.*parseInt*(st.nextToken());  String operation= st.nextToken();  **int** oprnd2= Integer.*parseInt*(st.nextToken());  // Perform the specified operation  **if**(operation.equals("+")) {  result= oprnd1+ oprnd2;  } **else** **if**(operation.equals("-")) {  result= oprnd1- oprnd2;  } **else** **if**(operation.equals("\*")) {  result= oprnd1\* oprnd2;  } **else**{  result= oprnd1/ oprnd2;  }  // Display a message and send the result back to the client  System.***out***.println("Sending the result...");  dos.writeUTF(Integer.*toString*(result));  }  }  } |

## OUTPUT (Screen Shots)

|  |  |
| --- | --- |
| #1 | 사칙연산(client에서) |
|  | |
| #2 | 서버에서 받은 값과 값 전송 |
|  | |