```java
import java.io.*;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.UUID;

public class server {
    public static void main(String[] args) {
        try {
            // Create a server socket listening on port 8080
            ServerSocket serverSocket = new ServerSocket(8080);
            System.out.println("Server is listening on port 8080...");

            // Wait for a client to connect
            Socket clientSocket = serverSocket.accept();
            System.out.println("Client connected.");

            // Create input stream for communication with the client
            BufferedReader in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));

            // Read the number of files to expect
            int numFiles = 20;

            System.out.println("Expecting " + numFiles + " files from the client.");

            // Receive files from the client
            for (int i = 0; i < numFiles; i++) {
                // Generate a random file name
                String fileName = UUID.randomUUID().toString() + ".txt";

                // Read the file content from the client
                StringBuilder fileContent = new
```

```
31 StringBuilder();
32                 String line;
33                 while ((line = in.readLine()) != null
    && !line.equals("END_OF_FILE")) {
34                     fileContent.append(line).append("
   \n");
35                 }
36 
37                 // Save the file content to a file
38                 try (FileOutputStream
   fileOutputStream = new FileOutputStream("received_"
    + fileName)) {
39                     fileOutputStream.write(
   fileContent.toString().getBytes());
40                     System.out.println("File content
   saved to received_" + fileName);
41                 } catch (IOException e) {
42                     System.err.println("Error saving
   file content to received_" + fileName);
43                     e.printStackTrace();
44                 }
45             }
46 
47             // Close the streams and sockets
48             in.close();
49             clientSocket.close();
50             serverSocket.close();
51 
52         } catch (IOException e) {
53             e.printStackTrace();
54         }
55     }
56 }
57 
```