TÖL401G Assignment 8

Helgi Grétar Gunnarsson, Rafnar Ólafsson

TOTAL POINTS

7.5 / 10

QUESTION 1

1 Assignment 8 7.5 / 10

- **0 pts** Correct

✓ - **0.5 pts** **There is no clear description of output**

- **1 pts** Server screenshot does not presented
- **1 pts** Second client screenshot does not presented

✓ - **1 pts** **There is no information from server**

- **1 pts** First client screenshot does not presented

✓ - **1 pts** **Presented results do not demonstrate that multithreading was used**

- **0 pts** No client

Fyrri hluti - Mynd tengist báðum forritum.

```
import java.net.*;
import java.io.*;
class Connection {
  DataInputStream in;
  DataOutputStream out;
  Socket clientSocket;
  public Connection(Socket aClientSocket) {
    try {
      clientSocket = aClientSocket;
      in = new DataInputStream(clientSocket.getInputStream());
      out = new DataOutputStream(clientSocket.getOutputStream());
      handleRequest();
    } catch (IOException e) { System.out.println("Connection:" + e.getMessage()); }
  }
  static long fibonacci(long count){
    if (count <= 1) return count;
    else return fibonacci(count - 1) + fibonacci(count - 2);
  }
  public void handleRequest() {
    try { // an echo server
        String data = in.readUTF(); // read a line of data from the stream
        long parsedData = Long.parseLong(data);
        out.writeUTF(Long.toString(fibonacci(parsedData)));
    } catch (EOFException e) { System.out.println("EOF:" + e.getMessage());
    } catch (IOException e) { System.out.println("readline:" + e.getMessage());
    } finally {
      try {
        clientSocket.close();
      } catch (IOException e) {/* close failed */
      }
    }
  }
}
```

```
import java.net.*;
import java.io.*;

public class ConnectionOrientedClient {
 public static void main(String args[]) {
  // args[0]: message contents, args[1]: destination hostname
  Socket aSocket = null;
```

```
   try {
    int serverPort = 7896;
    aSocket = new Socket(args[1], serverPort);
    DataInputStream in = new DataInputStream(aSocket.getInputStream());
    DataOutputStream out = new DataOutputStream(aSocket.getOutputStream());
    out.writeUTF(args[0]); // UTF is a string encoding
    String data = in.readUTF(); // read a line of data from the stream
    System.out.println("Received: " + data);
   } catch (UnknownHostException e) { System.out.println("Socket:" + e.getMessage());
   } catch (EOFException e) { System.out.println("EOF:" + e.getMessage());
   } catch (IOException e) { System.out.println("readline:" + e.getMessage());
   } finally {
    if (aSocket != null)
    try {
     aSocket.close();
    } catch (IOException e) { System.out.println("close:" + e.getMessage());
    }
   }
  }
}
```

```
import java.net.*;
import java.io.*;

public class ConnectionOrientedServer {
  public static void main(String args[]) {
   try {
    int serverPort = 7896; // the server port
    ServerSocket listenSocket = new ServerSocket(serverPort);
    while (true) {
     Socket clientSocket = listenSocket.accept();
     Connection c = new Connection(clientSocket); // Handle request
    }
   } catch (IOException e) {
    System.out.println("Listen socket:" + e.getMessage());
   }
  }
}
```

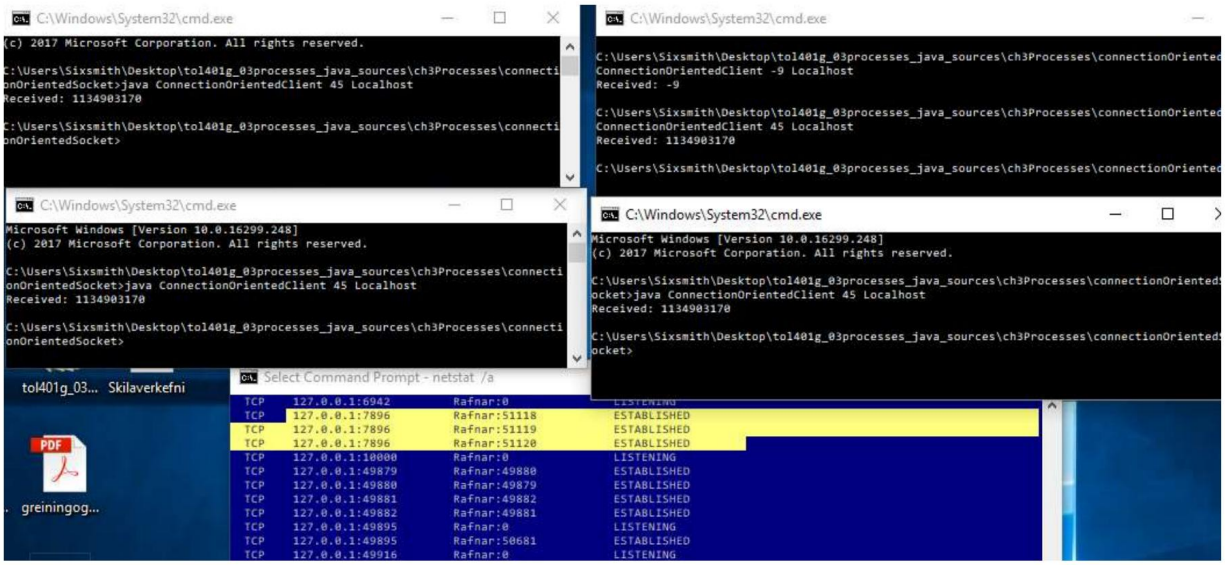
Seinni hluti

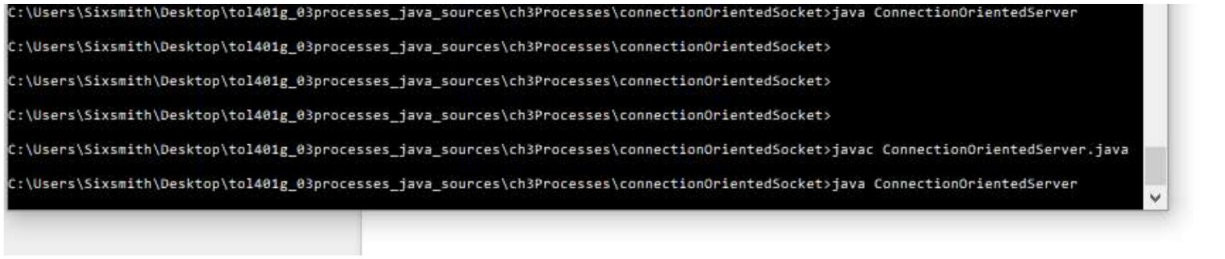
```
import java.net.*;
import java.io.*;
class Connection extends Thread
{
  DataInputStream in;
  DataOutputStream out;
  Socket clientSocket;
  public Connection(Socket aClientSocket) {
    try {
      clientSocket = aClientSocket;
      in = new DataInputStream(clientSocket.getInputStream());
      out = new DataOutputStream(clientSocket.getOutputStream());
      //handleRequest();
    } catch (IOException e) { System.out.println("Connection:" + e.getMessage()); }
  }
  static long fibonacci(long count){
    if (count <= 1) return count;
    else return fibonacci(count - 1) + fibonacci(count - 2);
  }
  public void run() {
    try { // an echo server
        String data = in.readUTF(); // read a line of data from the stream
        long parsedData = Long.parseLong(data);
        out.writeUTF(Long.toString(fibonacci(parsedData)));
    } catch (EOFException e) { System.out.println("EOF:" + e.getMessage());
    } catch (IOException e) { System.out.println("readline:" + e.getMessage());
    } finally {
      try {
        clientSocket.close();
      } catch (IOException e) {/* close failed */
      }
    }
  }
}
```

```
import java.net.*;
import java.io.*;

public class ConnectionOrientedClient {
 public static void main(String args[]) {
```

```
  // args[0]: message contents, args[1]: destination hostname
  Socket aSocket = null;
  try {
    int serverPort = 7896;
    aSocket = new Socket(args[1], serverPort);
    DataInputStream in = new DataInputStream(aSocket.getInputStream());
    DataOutputStream out = new DataOutputStream(aSocket.getOutputStream());
    out.writeUTF(args[0]); // UTF is a string encoding
    String data = in.readUTF(); // read a line of data from the stream
    System.out.println("Received: " + data);
  } catch (UnknownHostException e) { System.out.println("Socket:" + e.getMessage());
  } catch (EOFException e) { System.out.println("EOF:" + e.getMessage());
  } catch (IOException e) { System.out.println("readline:" + e.getMessage());
  } finally {
    if (aSocket != null)
    try {
      aSocket.close();
    } catch (IOException e) { System.out.println("close:" + e.getMessage());
    }
  }
 }
}
```

```
import java.net.*;
import java.io.*;

public class ConnectionOrientedServer {
 public static void main(String args[]) {
  try {
    int serverPort = 7896; // the server port
    ServerSocket listenSocket = new ServerSocket(serverPort);
    while (true) {
      Socket clientSocket = listenSocket.accept();
      Connection c = new Connection(clientSocket); // Handle request
        c.start();
    }
  } catch (IOException e) {
    System.out.println("Listen socket:" + e.getMessage());
  }
 }
}
```


C:\Windows\System32\cmd.exe
(c) 2017 Microsoft Corporation. All rights reserved.
C:\Users\Sixsmith\Desktop\tol401g_03processes_java_sources\ch3Processes\connectionOrientedSocket>java ConnectionOrientedClient 45 Localhost
Received: 1134903170
C:\Users\Sixsmith\Desktop\tol401g_03processes_java_sources\ch3Processes\connectionOrientedSocket>
C:\Windows\System32\cmd.exe
C:\Users\Sixsmith\Desktop\tol401g_03processes_java_sources\ch3Processes\connectionOrientedClient -9 Localhost
Received: -9
C:\Users\Sixsmith\Desktop\tol401g_03processes_java_sources\ch3Processes\connectionOrientedClient 45 Localhost
Received: 1134903170
C:\Users\Sixsmith\Desktop\tol401g_03processes_java_sources\ch3Processes\connectionOriented
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.16299.248]
(c) 2017 Microsoft Corporation. All rights reserved.
C:\Users\Sixsmith\Desktop\tol401g_03processes_java_sources\ch3Processes\connectionOrientedSocket>java ConnectionOrientedClient 45 Localhost
Received: 1134903170
C:\Users\Sixsmith\Desktop\tol401g_03processes_java_sources\ch3Processes\connectionOrientedSocket>
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.16299.248]
(c) 2017 Microsoft Corporation. All rights reserved.
C:\Users\Sixsmith\Desktop\tol401g_03processes_java_sources\ch3Processes\connectionOrientedSocket>java ConnectionOrientedClient 45 Localhost
Received: 1134903170
C:\Users\Sixsmith\Desktop\tol401g_03processes_java_sources\ch3Processes\connectionOrientedSocket>
tol401g_03... Skilaverkefni
PDF
greiningog...
Select Command Prompt - netstat /a
TCP 127.0.0.1:6942 Rafnar:0 LISTENING
TCP 127.0.0.1:7896 Rafnar:51118 ESTABLISHED
TCP 127.0.0.1:7896 Rafnar:51119 ESTABLISHED
TCP 127.0.0.1:7896 Rafnar:51120 ESTABLISHED
TCP 127.0.0.1:10000 Rafnar:0 LISTENING
TCP 127.0.0.1:49879 Rafnar:49880 ESTABLISHED
TCP 127.0.0.1:49880 Rafnar:49879 ESTABLISHED
TCP 127.0.0.1:49881 Rafnar:49882 ESTABLISHED
TCP 127.0.0.1:49882 Rafnar:49881 ESTABLISHED
TCP 127.0.0.1:49895 Rafnar:0 LISTENING
TCP 127.0.0.1:49895 Rafnar:50681 ESTABLISHED
TCP 127.0.0.1:49916 Rafnar:0 LISTENING


C:\Users\Sixsmith\Desktop\tol401g_03processes_java_sources\ch3Processes\connectionOrientedSocket>java ConnectionOrientedServer
C:\Users\Sixsmith\Desktop\tol401g_03processes_java_sources\ch3Processes\connectionOrientedSocket>
C:\Users\Sixsmith\Desktop\tol401g_03processes_java_sources\ch3Processes\connectionOrientedSocket>
C:\Users\Sixsmith\Desktop\tol401g_03processes_java_sources\ch3Processes\connectionOrientedSocket>
C:\Users\Sixsmith\Desktop\tol401g_03processes_java_sources\ch3Processes\connectionOrientedSocket>javac ConnectionOrientedServer.java
C:\Users\Sixsmith\Desktop\tol401g_03processes_java_sources\ch3Processes\connectionOrientedSocket>java ConnectionOrientedServer

1 Assignment 8 7.5 / 10

- **0 pts** Correct

✓ **- 0.5 pts** **There is no clear description of output**

- **1 pts** Server screenshot does not presented
- **1 pts** Second client screenshot does not presented

✓ **- 1 pts** **There is no information from server**

- **1 pts** First client screenshot does not presented

✓ **- 1 pts** **Presented results do not demonstrate that multithreading was used**

- **0 pts** No client