

CN2 Aufgabe 2

Julian Bertol

October 24, 2024

1 Aufgabe 1

1.1 erbinden Sie sich mit telnet zu einem Server unter `www.google.de` auf Port 80. Nutzen Sie das in der Vorlesung gewonnenen Wissen, um die Ressource „/“ anzuzeigen. Bedenken Sie, dass Sie eingegebene Zeichen eventuell nicht sehen. Sie werden dennoch gesendet. Falls Sie nicht schnell genug tippen können, bereiten Sie Ihre Anfrage in einem Texteditor vor, kopieren sie in die Zwischenablage und fügen die Anfrage dann in putty oder telnet ein.

- GET / http/1.1
Host: www.google.de

1.2 Sehen Sie eine Antwort vom Server?

- ja, http/1.1 200 ok

1.3 Wie können Sie den Server dazu bringen die Verbindung am Ende zu schließen? Wie können Sie dem Server mitteilen, dass die Verbindung offengelassen werden soll?

- Connection: open
- Connection: close

```
HTTP/1.1 200 OK
Date: Thu, 24 Oct 2024 12:24:43 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=ISO-8859-1
Content-Security-Policy-Report-Only: object-src 'none';base-uri 'self';script-src 'nonce-Zrsw2OM3W4wSH-nyO4RS_Q' 'strict-dynamic' 'report-sample' 'unsafe-eval' 'unsafe-inline' https: http:;report-uri https://csp.withgoogle.com/csp/gws/other
Server: gws
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN
Set-Cookie: AEC=AVYB7crgkFYsNQMYM9KhW3xYqADf8lnShCo3JL6ugIaGVtDfEHevl5GVPyk; expires=Tue, 22-Apr-2025 12:24:43 GMT; path=/; domain=.google.de; Secure; HttpOnly; SameSite=lax
Accept-Ranges: none
Vary: Accept-Encoding
Connection: close
Transfer-Encoding: chunked
```

1.4 Wurde der Antwort auch ein Cookie beigefügt?, Wie lautete der Inhalt des Cookies?

- ja, Set-Cookie: AEC=AVYB7crgkFYsNQMYM9KhW3xYqADf81nShCo3JL6ugIaGVtDfEHv15GVPyk;

1.5 Wie lange ist der/das Cookie gültig?

- expires=Tue, 22-Apr-2025 12:24:43 GMT
- Bis zum 22. April 2025

1.6 Haben Sie eine Vermutung, was das Cookie sein könnte?

- Das Cookie könnte beispielsweise dazu verwendet werden, um den Nutzer zu identifizieren und Sitzungsinformationen zu speichern.

2 Aufgabe 2

2.1 MyHttpClient Klasse

```
1 public class MyHttpClient {
2     int meineWelt = 0;
3     public static void main(String[] args) {
4         MyHttpClient meineEigeneVariable = new MyHttpClient(1);
5         String einText = "Hallo";
6         meineEigeneVariable.printHalloWelt(einText, "Welt");
7     }
8     public MyHttpClient(int zahl) {
9         // dies ist der Konstruktor
10        System.out.println("Ich wurde erstellt!");
11        meineWelt = zahl;
12    }
13    public void printHalloWelt(String text1, String text2) {
14        System.out.println(text1 + text2 + meineWelt);
15    }
16 }
```

2.2 Was gibt das Programm aus, wenn Sie die main-Methode ändern

```
1 public static void main(String[] args) {
2     MyHttpClient meineEigeneVariable = new MyHttpClient(1);
3     MyHttpClient meineZweiteVariable = new MyHttpClient(2);
4     String einText = "Hallo";
5     meineEigeneVariable.printHalloWelt(einText, "Welt");
6     meineZweiteVariable.printHalloWelt(einText, "Welt");
7 }
```

- Ausgabe
 - Ich wurde erstellt
 - ich wurde erstellt
 - HalloWelt1
 - HalloWelt2

2.3 Was gibt unser Programm aus, wenn wir unserer main-Methode erweitern?

```
1 public class MyHttpClient {
2     int meineWelt = 0;
3     public static void main(String[] args) {
4         MyHttpClient meineEigeneVariable = new MyHttpClient(1);
5         MyHttpClient meineZweiteVariable = new MyHttpClient(2);
6         String einText = "Hallo";
7         meineEigeneVariable.printHalloWelt(einText, "Welt");
8         meineZweiteVariable.printHalloWelt(einText, "Welt");
9         MyHttpClient meineDritteVariable = new MyHttpClient(1337);
10        int zahl = meineDritteVariable.whichWorld();
11        System.out.println("Die dritte erstellte Variable ist die "+zahl+" Welt");
12    }
13    public MyHttpClient(int zahl) {
14        // dies ist der Konstruktor
15        System.out.println("Ich wurde erstellt!");
16        meineWelt = zahl;
17    }
18    public void printHalloWelt(String text1, String text2) {
19        System.out.println(text1 + text2 + meineWelt);
20    }
21    public int whichWorld() {
22        return meineWelt;
23    }
24 }
```

- Ausgabe
 - Ich wurde erstellt!
 - Ich wurde erstellt!
 - HalloWelt1
 - HalloWelt2
 - Ich wurde erstellt!
 - Die dritte erstellte Variable ist die 1337. Welt

2.4 Die eigentliche Aufgabe

```
1 package org.example;
2
3 import java.io.*;
4 import java.net.InetAddress;
5 import java.net.Socket;
6 import java.net.UnknownHostException;
7
8 public class MyHttpClient {
9
10     public static InetAddress setAddress(String url) throws UnknownHostException {
11         InetAddress address = InetAddress.getByName(url);
12         System.out.println("IP-Adresse f r " + url + ": " + address);
13         return address;
14     }
15
16     public static Socket createSocket(InetAddress address, int port) throws
17         IOException {
18         Socket socket = new Socket(address, port);
19         System.out.println("Socket erstellt: " + socket);
20         return socket;
21     }
22
23     public static void sendmsg(Socket socket) throws IOException {
24         OutputStream os = socket.getOutputStream();
25         PrintWriter pw = new PrintWriter(os, true); //AutoFlush, ohne das funktioniert
26         es nicht
27
28         pw.println("GET / HTTP/1.1");
29         pw.println("Host: " + socket.getInetAddress().getHostName());
30         pw.println("Connection: close");
31         pw.println();
32         pw.println();
33     }
34
35     public static String receivemsg(Socket socket) throws IOException {
36         BufferedReader br = new BufferedReader(new InputStreamReader(socket.
37             getInputStream()));
38         String str;
39         StringBuilder answer = new StringBuilder();
40
41         while ((str = br.readLine()) != null) {
42             answer.append(str);
43             answer.append(System.lineSeparator()); //sorgt daf r , das untereinander
44             geschrieben wird
45         }
46
47         System.out.println(answer.toString());
48
49         return answer.toString();
50     }
51
52     public static Boolean getcookies(String answer){
53         CharSequence setCookieString = "Set-Cookie";
54         Boolean found = false;
55         for (String singleLine : answer.split("\\r?\\n")) {
56             if (singleLine.contains(setCookieString)) {
57                 System.out.println("Cookie-Zeile: " + singleLine);
58                 found = true;
59             }
60         }
61
62         return found;
63     }
64
65     public static void main(String[] args) {
66         InetAddress test = null;
67         String answer = null;
68         try {
```

```

66         test = setAddress("www.google.de");
67     } catch (UnknownHostException e) {
68         System.err.println("Unbekannter Host: " + e.getMessage());
69         return;
70     }
71
72     try (Socket socket = createSocket(test, 80)) {
73         sendmsg(socket);
74         answer = receivemsg(socket);
75     } catch (IOException e) {
76         System.err.println("IOException: " + e.getMessage());
77     }
78
79     Boolean cookies = getcookies(answer);
80     if(!cookies)
81         System.out.println("Cookies nicht gefunden");
82 }
83 }

```