

Template Week 6 – Networking

Bonus point assignment – week 6

Remember that bitwise java application you've made in week 2? Expand that application so that you can also calculate a network segment as explained in the PowerPoint slides of week 6. Use the bitwise & AND operator. You need to be able to input two Strings. An IP address and a subnet.

IP: 192.168.1.100 and subnet: 255.255.255.224 for /27

Example: 192.168.1.100/27

Calculate the network segment

IP Address: 11000000.10101000.00000001.01100100

Subnet Mask: 11111111.11111111.11111111.11100000

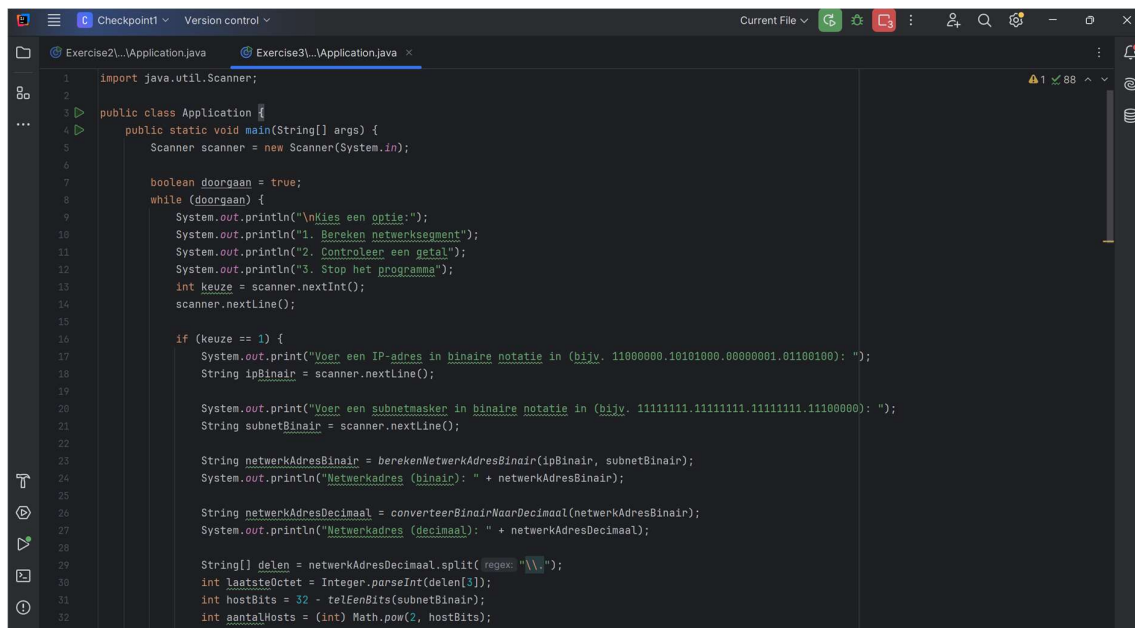
Network Addr: 11000000.10101000.00000001.01100000

This gives 192.168.1.96 in decimal as the network address.

For a /27 subnet, each segment (or subnet) has 32 IP addresses (2^5).

The range of this network segment is from 192.168.1.96 to 192.168.1.127.

Paste source code here, with a screenshot of a working application.



```
1 import java.util.Scanner;
2
3 public class Application {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6
7         boolean doorgaan = true;
8         while (doorgaan) {
9             System.out.println("\nKies een optie:");
10            System.out.println("1. Bereken netwerksegment");
11            System.out.println("2. Controleer een getal");
12            System.out.println("3. Stop het programma");
13            int keuze = scanner.nextInt();
14            scanner.nextLine();
15
16            if (keuze == 1) {
17                System.out.print("Voer een IP-adres in binaire notatie in (bijv. 11000000.10101000.00000001.01100100): ");
18                String ipBinair = scanner.nextLine();
19
20                System.out.print("Voer een subnetmasker in binaire notatie in (bijv. 11111111.11111111.11111111.11100000): ");
21                String subnetBinair = scanner.nextLine();
22
23                String netwerkAdresBinair = berekenNetwerkAdresBinair(ipBinair, subnetBinair);
24                System.out.println("Netwerkadres (binair): " + netwerkAdresBinair);
25
26                String netwerkAdresDecimaal = converteerBinairNaarDecimaal(netwerkAdresBinair);
27                System.out.println("Netwerkadres (decimaal): " + netwerkAdresDecimaal);
28
29                String[] delen = netwerkAdresDecimaal.split(" ");
30                int laatsteOctet = Integer.parseInt(delen[3]);
31                int hostBits = 32 - tellenBits(subnetBinair);
32                int aantalHosts = (int) Math.pow(2, hostBits);
```

```
33
34
35 String beginIP = netwerkAdresDecimaal;
36 String eindIP = delen[0] + "." + delen[1] + "." + delen[2] + "." + (laatsteOctet + aantalHosts - 1);
37
38 System.out.println("IP-bereik: " + beginIP + " tot " + eindIP);
39 } else if (keuze == 2) {
40     System.out.print("Voer een heel getal in: ");
41     int getal = scanner.nextInt();
42
43     System.out.println("1. Is het getal oneven?");
44     System.out.println("2. Is het getal een macht van 2?");
45     System.out.println("3. Wat is het twee's complement?");
46     int subKeuze = scanner.nextInt();
47
48     if (subKeuze == 1) {
49         if (getal % 2 != 0) {
50             System.out.println("Het getal is oneven.");
51         } else {
52             System.out.println("Het getal is even.");
53         }
54     } else if (subKeuze == 2) {
55         if (getal > 0 && (getal & (getal - 1)) == 0) {
56             System.out.println("Het getal is een macht van 2.");
57         } else {
58             System.out.println("Het getal is geen macht van 2.");
59         }
60     } else if (subKeuze == 3) {
61         System.out.println("Twee's complement: " + (~getal + 1));
62     } else {
63
64     }
65 }
66
67 scanner.close();
68
69 }
70
71
72
73
74
75 @
76 private static String berekenNetwerkAdresBinair(String ip, String subnet) { 1 usage
77     String[] ipDelen = ip.split(regex: "\\.");
78     String[] subnetDelen = subnet.split(regex: "\\.");
79     StringBuilder netwerkAdres = new StringBuilder();
80
81     for (int i = 0; i < 4; i++) {
82         int ipDeel = Integer.parseInt(ipDelen[i], radix: 2);
83         int subnetDeel = Integer.parseInt(subnetDelen[i], radix: 2);
84         int netwerkDeel = ipDeel & subnetDeel;
85         netwerkAdres.append(String.format("%8s", Integer.toBinaryString(netwerkDeel)).replace(' ', '0'));
86         if (i < 3) {
87             netwerkAdres.append(".");
88         }
89     }
90 }
```

```
87     }
88 }
89
90 return netwerkAdres.toString();
91 }
92
93 @private static int telEnBits(String subnet) { 1 usage
94     String[] subnetDelen = subnet.split(regex: "\\.");
95     int count = 0;
96     for (String deel : subnetDelen) {
97         count += Integer.bitCount(Integer.parseInt(deel, radix: 2));
98     }
99     return count;
100 }
101
102 @private static String converteerBinairNaarDecimaal(String binairAdres) { 1 usage
103     String[] delen = binairAdres.split(regex: "\\.");
104     StringBuilder decimaalAdres = new StringBuilder();
105
106     for (int i = 0; i < 4; i++) {
107         int decimaalDeel = Integer.parseInt(delen[i], radix: 2);
108         decimaalAdres.append(decimaalDeel);
109         if (i < 3) {
110             decimaalAdres.append(".");
111         }
112     }
113
114     return decimaalAdres.toString();
115 }
116 }
117
118 }
119
120     decimaalAdres.append(".");
121 }
122 }
123
124     return decimaalAdres.toString();
125 }
126 }
127
128 }
```

Run Application Application Application

1. Bereken netwerksegment
2. Controleer een getal
3. Stop het programma

Voer een IP-adres in binaire notatie in (bijv. 11000000.10101000.00000001.01100100): 11000000.10101000.00000001.01100100
Voer een subnetmasker in binaire notatie in (bijv. 11111111.11111111.11111111.11100000): 11111111.11111111.11111111.11100000
Netwerkadres (binair): 11000000.10101000.00000001.01100100
Netwerkadres (decimaal): 192.168.1.96
IP-bereik: 192.168.1.96 tot 192.168.1.127

Kies een optie:
1. Bereken netwerksegment
2. Controleer een getal
3. Stop het programma

Ready? Save this file and export it as a pdf file with the name: [week6.pdf](#)