Template Week 6 – Networking

Student number: 572121

Assignment 6.1: Working from home

Screenshot installation openssh-server:

Screenshot successful SSH command execution:

Screenshot successful execution SCP command:

Screenshot remmina:

Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:

Screenshot website visit via IP address:

Assignment 6.3: subnetting

How many IP addresses are in this network configuration 192.168.110.128/25?

What is the usable IP range to hand out to the connected computers?

Check your two previous answers with this calculator:
https://www.calculator.net/ip-subnet-calculator.html

Explain the above calculation in your own words.

Assignment 6.4: HTML

Screenshot IP address Ubuntu VM:

Screenshot of Site directory contents:

Screenshot python3 webserver command:

Screenshot web browser visits your site

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 (2⁵).
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.

```
import nl.saxion.app.SaxionApp;

public class Application implements Runnable {
    public static void main(String[] args) {
        SaxionApp.start(new Application(), 1024, 768);
    }

    public void run() {
        // laat menu opties zien
        SaxionApp.printLine("Keuzemenu: 572121");
        SaxionApp.printLine("1. Is number odd?");
        SaxionApp.printLine("2. Is number a power of 2?");
        SaxionApp.printLine("3. Two's complement of number");
        SaxionApp.printLine("4. Calculate network segment");

        // lees keuze van gebruiker
        int choice = SaxionApp.readInt();

        // voer gekozen optie uit
        if (choice == 1) {
            checkOddEven();
        } else if (choice == 2) {
            checkPowerOf2();
        } else if (choice == 3) {
            calculateTwosComplement();
        } else if (choice == 4) {
            calculateNetworkSegment();
        } else {
            SaxionApp.printLine("Invalid choice!");
        }
    }

    // optie 1: kijk of nummer even of oneven is
    public void checkOddEven() {
        SaxionApp.printLine("Enter a number:");
        int number = SaxionApp.readInt();

        // simpele check: als delen door 2 een rest heeft is het oneven
        if (number % 2 == 0) {
            SaxionApp.printLine(number + " is even");
        } else {
            SaxionApp.printLine(number + " is odd");
        }
    }

    // optie 2: kijk of nummer een macht van 2 is
    public void checkPowerOf2() {
        SaxionApp.printLine("Enter a number:");
```

```
        int number = SaxionApp.readInt();

        // een nummer is een macht van 2 als het maar één '1' bit heeft
        boolean isPowerOf2 = number > 0 && (number & (number - 1)) == 0;

        if (isPowerOf2) {
            SaxionApp.printLine(number + " is a power of 2");
        } else {
            SaxionApp.printLine(number + " is not a power of 2");
        }
    }

    // optie 3: bereken twee-complement
    public void calculateTwosComplement() {
        SaxionApp.printLine("Enter a number:");
        int number = SaxionApp.readInt();

        // bereken twee-complement
        int twosComplement = ~number + 1;

        SaxionApp.printLine("Two's complement of " + number + " is: " + twosComplement);

        // laat zien dat we terug kunnen naar het originele nummer
        int backToOriginal = ~twosComplement + 1;
        SaxionApp.printLine("Converting back gives us: " + backToOriginal);
    }

    // optie 4: bereken netwerk segment
    public void calculateNetworkSegment() {
        // vraag ip adres van gebruiker
        SaxionApp.printLine("Enter IP address:");
        String userIpAddress = SaxionApp.readString();

        // vraag subnet mask van gebruiker
        SaxionApp.printLine("Enter subnet mask:");
        String userSubnetMask = SaxionApp.readString();

        // splits ip adres in vier delen
        String[] ipParts = userIpAddress.split("\\.");

        // splits subnet mask in vier delen
        String[] subnetParts = userSubnetMask.split("\\.");

        // zet elk deel om naar een nummer
        int[] ipNumbers = new int[4];
        int[] subnetNumbers = new int[4];

        for (int i = 0; i < 4; i++) {
```

```
            ipNumbers[i] = Integer.parseInt(ipParts[i]);
            subnetNumbers[i] = Integer.parseInt(subnetParts[i]);
        }

        // bereken netwerk adres met en-operatie (&)
        int[] networkAddress = new int[4];
        for (int i = 0; i < 4; i++) {
            networkAddress[i] = ipNumbers[i] & subnetNumbers[i];
        }

        // toon ip adres in binair
        SaxionApp.printLine("\nIP Address in binary:");
        displayinbinary(ipNumbers);

        // toon subnet mask in binair
        SaxionApp.printLine("\nSubnet Mask in binary:");
        displayinbinary(subnetNumbers);

        // toon een scheidingslijn
        SaxionApp.printLine("----------------------------------------");

        // toon netwerk adres in binair
        SaxionApp.printLine("\nNetwork Address in binary:");
        displayinbinary(networkAddress);

        // toon netwerk adres in decimaal formaat
        SaxionApp.printLine("\nNetwork Address in decimal format:");
        SaxionApp.printLine(networkAddress[0] + "." +
                networkAddress[1] + "." +
                networkAddress[2] + "." +
                networkAddress[3]);
    }

    // hulp functie om getallen in binair formaat te tonen (8 cijfers elk)
    private void displayinbinary(int[] numbers) {
        // ga door elk getal heen (van numbers[0] tot numbers[3])
        for (int i = 0; i < 4; i++) {
            // pak het huidige getal dat we willen omzetten
            int currentnumber = numbers[i];

            // maak een lege string voor ons binaire resultaat
            String binarynumber = "";

            // zet om naar binair door herhaaldelijk te delen door 2
            // en het controleren van de rest (0 of 1)
            int remainingnumber = currentnumber;
            for (int bit = 0; bit < 8; bit++) {  // we hebben 8 bits nodig
                // krijg het volgende binaire cijfer (0 of 1)
```

```
            int remainder = remainingnumber % 2;
            // voeg het toe aan de voorkant van ons resultaat
            binarynumber = remainder + binarynumber;
            // deel het getal door 2 voor de volgende stap
            remainingnumber = remainingnumber / 2;
        }

        // print het binaire getal
        SaxionApp.print(binarynumber);

        // voeg een punt toe na het getal (behalve voor de laatste)
        if (i < 3) {
            SaxionApp.print(".");
        }
    }
  }
}
```

Saxion Drawingboard

```
Keuzemenu: 572121
1. Is number odd?
2. Is number a power of 2?
3. Two's complement of number
4. Calculate network segment
4
Enter IP address:
192.168.1.100
Enter subnet mask:
255.255.255.224

IP Address in binary:
11000000.10101000.00000001.01100100
Subnet Mask in binary:
11111111.11111111.11111111.11100000----------

Network Address in binary:
11000000.10101000.00000001.01100000
Network Address in decimal format:
192.168.1.96
```

Ready? Save this file and export it as a pdf file with the name: **week6.pdf**