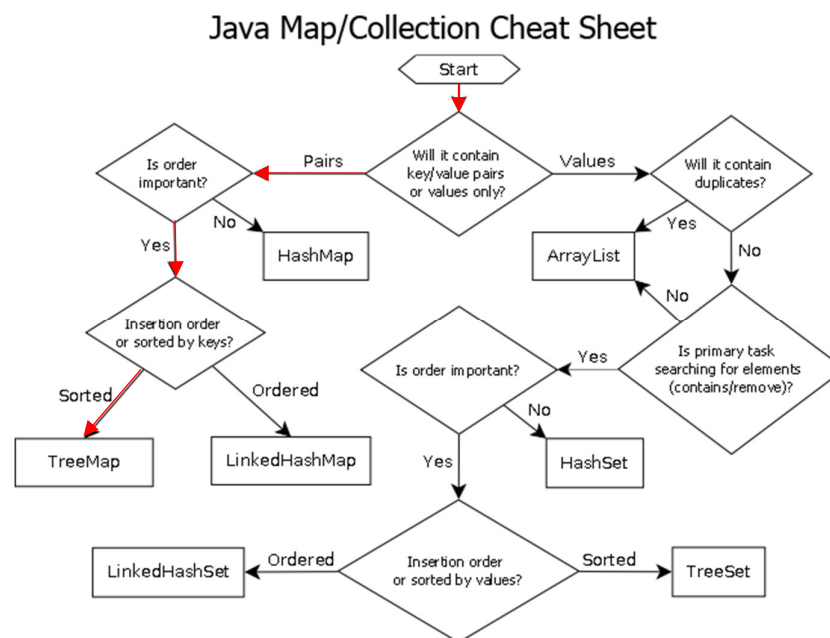


Opgave 7

Keuze van datastructuur

De periodes heb ik in een ArrayList geplaatst, maar hoe de vakken gelinkt worden met hen waarden heb ik als volgt gevonden:



Op de eerste vraag is het antwoord pairs, omdat we vakken met hun aantal inschrijvingen moeten linken. Op de tweede vraag antwoordde ik ja, omdat we willen dat elke periodes de vakken in de zelfde volgorde wordt afgedrukt. Bij onze laatste vraag koos ik voor sorted omdat we het willen sorteren volgende de namen van de vakken. We komen dus uit dat we best een TreeMap gebruiken.

Programma

We starten met de klasse `Inschrijvingen`, hierin maak in een `ArrayList` aan genaamd `periodeArray` met daarin een `TreeMap` die een `String` in `Integer` bewaard.

Daarna vul ik de `ArrayList` met lege `TreeMaps`, dit is nodig voor daarna de `TreeMaps` in te vullen.

```
public class Inschrijvingen
{
    ArrayList<TreeMap<String,Integer>> periodeArray;

    public Inschrijvingen()
    {
        this.periodeArray = new ArrayList();
        for (int i = 0; i < 4; i++) {
            this.periodeArray.add(new TreeMap<>());
        }
    }
}
```

De method zal het bestand inlezen en de gegeven bewerken.

```
public void read (String pathname) throws IOException
{
    for (String line: Files.readAllLines(Path.of(pathname)))
    {
        String[] dataString = line.split( regex: " ");

        String vak = dataString[2];

        String periodeString = dataString[3];
        int tempPeriode = Integer.parseInt(periodeString);

        TreeMap<String,Integer> map = periodeArray.get(tempPeriode-1);

        if(!map.containsKey(vak))
            map.put(vak,1);
        else
        {
            int value = map.get(vak);
            value ++;
            map.put(vak, value);
        }
    }
}
```

Eerst gaan we de file openen en elke regel splitsen in verschillende delen. We gebruiken enkel het derde deel uit de zin die ons het vak verteld en het vierde

die ons zegt in welke periode de persoon dit vak volgt. Dit getal zetten we ook naar een integer.

We gaan naar de TreeMap die zich bevindt in de juiste periode aanpassen. We kijken met een if statement of het vak al voorkomt. Is dit niet het geval dan steken met dit vak in de map samen met value 1 omdat dit de eerste inschrijving is. Komt het vak al wel voor halen we het aantal inschrijvingen op, tellen er 1 bij op en zetten het teug in de TreeMap.

Volgende methode zal alle gegevens afprinten.

```
public void print()
{
    for (int i = 0; i < periodeArray.size(); i++)
    {
        System.out.println("Periode "+(i+1));
        TreeMap<String,Integer> map = periodeArray.get(i);
        for (Map.Entry<String, Integer> entry : map.entrySet() )
        {
            System.out.println("        "+entry.getKey()+" "+entry.getValue());
        }
    }
}
```

Met een for loop gaan we alle periodes af, we halen hier ook weer de TreeMap op die overeenkomt met de periode. In de for loop doen we een for each loop die over alle entries van de map gaat overlopen. We printen hierbij de Key (vak) af gevolgd door de Value (aantal inschrijvingen).

In de main maken we een object Inschrijvingen aan, we vullen deze met het bestandje gegevens.txt en printen alles af.

```
public class Main {

    public static void main(String[] args) {programTest();}

    public static void programTest()
    {
        try
        {
            Inschrijvingen insch = new Inschrijvingen();
            insch.read( pathname: "D:\\Documenten\\UAssem3\\Java\\3-Datastructures\\Sessie7\\gegevens.txt");
            insch.print();
        } catch (IOException e)
        {
            System.out.println(e.getMessage());
        }
    }
}
```

Als resultaat van het afdrucken bekomen we het volgende:

Periode 1		
Netwerken		4
Software		3
Wiskunde		2
Periode 2		
Netwerken		2
Software		2
Wiskunde		3
Periode 3		
Netwerken		3
Wiskunde		3
Periode 4		
Software		4
Wiskunde		1