

P2W5 Opdracht Frequentie

C Data		
i	MAX	int
i	generator	Random
i	getallen	List<Integer>
m	Data()	
m	getGetallen()	List<Integer>

C Frequentie		
i	MAX	int
i	lijst	List<Integer>
i	frequenties	int[]
m	Frequentie(List<Integer>)	
m	bepaalFrequenties()	void
m	toonFrequenties()	String
m	toString()	String

De Klasse Data

De klasse `Data` bevat een constante `MAX` met de waarde 100.

Als tweede static attribuut is een `Random` object met de naam `generator` opgenomen.

Tenslotte is er het attribuut `getallen` van het type `List<Integer>`.

- Voorzie het nodige in de *constructor*. Zorg ervoor dat er een `ArrayList` van `int` gemaakt wordt die met `MAX` willekeurige waarden in de range 0..9 gevuld wordt.
- Voorzie een getter `getGetallen` die de `List` teruggeeft.

De klasse Frequentie

De klasse `Frequentie` bevat een `int` constante `MAX` met de waarde 10, een `List<Integer>` met de naam `lijst` en een gewone tabel van `int` met de naam `frequenties`.

- Voorzie een constructor die een nieuwe `ArrayList` maakt op basis van de `List` die als parameter meegegeven wordt.
- Voorzie een *private* methode `bepaalFrequenties` waarin je voor elk van de `int`'s in de `lijst` bepaalt hoeveel keer ze voorkomen.
- Voorzie een methode `toonFrequenties` om de frequenties te tonen, roep hierin de vorige methode op.
- Voorzie ook een `toString`-methode om de gehele inhoud van de lijst met 10 waarden per regel te kunnen afdrukken.

De klasse TestFrequentie

Gebruik deze klasse om je programma te testen.

```
public class TestFrequentie {  
    public static void main(String[] args) {  
  
        Data data = new Data();  
        Frequentie frequentie = new Frequentie(data.getGetallen());  
  
        System.out.println("Inhoud tabel:\n" + frequentie);  
        System.out.println("Frequenties:\n" + frequentie.toonFrequenties());  
    }  
}
```

Mogelijke uitvoer:

Inhoud tabel:

```
3 0 8 5 9 5 6 9 7 9  
8 1 6 9 1 0 7 9 5 6  
4 7 3 5 9 9 3 1 7 9  
0 1 3 2 0 1 1 6 5 6  
7 5 6 2 7 1 2 7 3 9  
2 5 8 4 3 4 3 8 5 6  
0 8 2 3 3 6 4 0 2 6  
0 4 6 6 3 6 2 8 0 7  
1 9 0 5 1 4 8 0 4 4  
9 0 6 8 3 3 4 6 9 6
```

Frequenties:

```
0 --> 11  
1 --> 9  
2 --> 7  
3 --> 12  
4 --> 9  
5 --> 9  
6 --> 15  
7 --> 8  
8 --> 8  
9 --> 12
```