HO
GENT

H5 ArrayList en arrays - Oefeningen

Table of Contents

1. Doelstellingen

- Oefening 2.1: leren werken met een (Array)List
- Oefening 2.2 & 2.3: leren werken met 1-dim arrays van primitieve datatypes
- Oefening 2.4: leren werken met 1-dim arrays van objecten en 1- en 2-dim arrays van primitieve datatypes

2. Oefeningen

2.1. Eigenschappen

Schrijf een applicatie om karaktereigenschappen bij te houden in 2 lijsten: eentje met positieve en eentje met negatieve eigenschappen.

We stellen 3 keer (moet gemakkelijk wijzigbaar zijn naar meer of minder) de vraag aan de gebruiker om een karaktereigenschap in te voeren. Hierbij wordt telkens eerst gevraagd of dit een positieve of negatieve eigenschap is adhv een code (1 voor positief, 2 voor negatief). Indien de gebruiker iets anders dan 1 of 2 invoert, wordt dit opnieuw gevraagd. Daarna wordt een omschrijving van de eigenschap gevraagd. De eigenschap wordt dan telkens aan de juiste lijst toegevoegd.

Daarna worden de eigenschappen getoond: eerst de positieve en daarna de negatieve. Tussen 2 eigenschappen wordt een komma getoond en als er geen positieve of negatieve eigenschappen zijn, wordt dit eveneens aangegeven.

Voorbeelduitvoer:

```
Is het een positieve (=1) of negatieve (=2) eigenschap?
4
Is het een positieve (=1) of negatieve (=2) eigenschap?
1
Omschrijf deze eigenschap:
geduldig
Is het een positieve (=1) of negatieve (=2) eigenschap?
2
Omschrijf deze eigenschap:
koppig
Is het een positieve (=1) of negatieve (=2) eigenschap?
0
Is het een positieve (=1) of negatieve (=2) eigenschap?
1
Omschrijf deze eigenschap:
ordentelijk
Positieve eigenschappen: geduldig, ordentelijk
Negatieve eigenschappen: koppig
```

OF

```
Is het een positieve (=1) of negatieve (=2) eigenschap?
1
Omschrijf deze eigenschap:
attent
Is het een positieve (=1) of negatieve (=2) eigenschap?
1
Omschrijf deze eigenschap:
vastberaden
Is het een positieve (=1) of negatieve (=2) eigenschap?
1
Omschrijf deze eigenschap:
leergierig
Positieve eigenschappen: attent, vastberaden, leergierig
Geen negatieve eigenschappen
```

2.2. Tekening

Schrijf een programma dat een tekening maakt op basis van een array.

In de array vind je ofwel een positief getal, een negatief getal of een 0. Een positief getal stelt een aantal sterretjes voor, de absolute waarde van een negatief getal stelt een aantal spaties voor en een 0 is een newline.

Teken volgende array:

```
int[] tekening = {52,0,23,-3,26,0,23,-6,23,0,23,-7,22,0,23,-7,22,0,22,-8,22,0,21,-9,
22,0,20,-23,9,0,18,-26,8,0,7,-8,2,-26,9,0,7,-8,2,-25,10,0,7,-8,2,-27,8,0,7,-8,2,-26,9
,0,7,-8,2,-24,11,0,7,-8,2,-25,10,0,7,-8,2,-24,11,0,7,-8,2,-23,12,0,7,-8,2,-22,13,0,7,-
8,2,-21,14,0,7,-8,37,0,52,0};
```

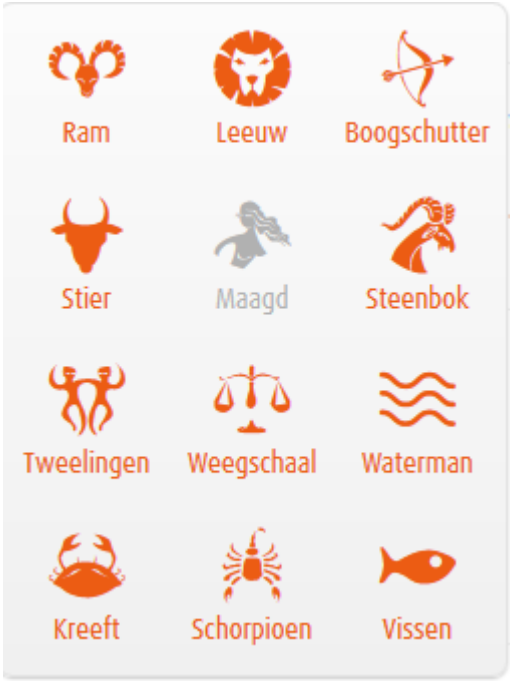
Maak voor de uitvoer gebruik van `System.out.print("*")`, `print(" ")` en `println()`.

2.3. Euromillions

Schrijf een applicatie die een trekking van Euromillions simuleert:

```
De getallen zijn [4, 13, 25, 46, 48]
De sterren [7, 10]
Het sterrenbeeld = Schorpioen
```

Er worden 5 **verschillende** cijfers gegenereerd uit het interval [1,50], 2 **verschillende** sterren uit het interval [1,12] en een sterrenbeeld.



Maak gebruik van 3 arrays:

- één waar de 5 getrokken getallen moeten inkomen
- één waar de 2 getrokken sterren moeten inkomen en
- één die de informatie over de beschikbare sterrenbeelden bijhoudt.

Schrijf je code zo efficiënt mogelijk en zorg ervoor dat indien het aantal getallen en/of sterren, of de bovengrenzen zouden wijzigen, er minimale aanpassingen aan de code moeten gebeuren.

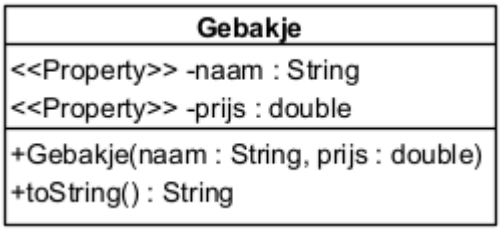
Tips:

- Om de arrays voor de getallen en de sterren te sorteren, hoef je geen eigen code te schrijven. Je kan hiervoor gebruik maken van de methode sort uit de klasse Arrays.
  Bijvoorbeeld: `Arrays.sort(mijnArray);`
- Ook om een array te tonen zoals in de voorbeelduitvoer (met vierkante haakjes er rond en met komma's tussen de verschillende elementen), bestaat er een methode in de klasse Arrays, namelijk toString. Je gebruikt die op dezelfde manier als de sort-methode: `System.out.println(Arrays.toString(mijnArray));`

2.4. Gebakjes

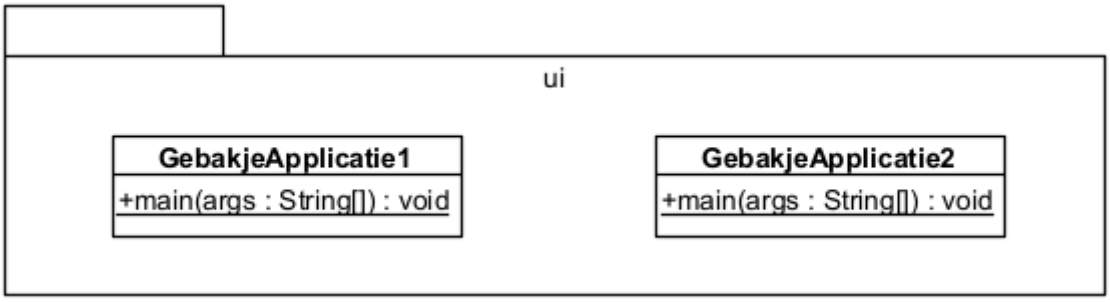
We helpen onze plaatselijke bakker met de administratie van zijn bestellingen van gebakjes.

Van een gebakje houden we de naam en prijs bij. Hiervoor is reeds een domeinklasse gemaakt met volgende UML:



We schrijven nu zelf 2 applicaties:

1. We nemen de bestelling op voor één klant en berekenen het bedrag dat de klant hiervoor moet betalen.
2. We nemen de bestelling op voor 3 klanten en maken een overzicht van het aantal te bakken gebakjes per soort.



Applicatie 1:

Het assortiment van de bakker bestaat uit het volgende:

String[] namen = {"aardbeientaartje", "eclair", "miserable", "hoorntje"};
double[] eenheidsprijzen = {2.75, 1.5, 2, 1.75};

Maak gebruik van de bovenstaande arrays en bouw zo de array gebakjes op die dus 4 Gebakje-objecten zal bevatten ⇒ aardbeientaartje 2.75, eclair 1.5, …

Laat de klant daarna van elk gebakje invoeren hoeveel stuks hij er van wil en hou deze aantallen bij in een aparte array bestelling.

Zorg tenslotte voor de gepaste uitvoer. Een gebakje dat niet besteld wordt, wordt niet vermeld in het overzicht van de bestelling met bedrag.

Voorbeeld van mogelijke uitvoer:

```
Assortiment gebakjes: [aardbeientaartje (€2,75), eclair (€1,50), miserable (€2,00), hoorntje (€1,75)]

Geef aantal voor aardbeientaartje (€2,75): 2
Geef aantal voor eclair (€1,50): 0
Geef aantal voor miserable (€2,00): 4
Geef aantal voor hoorntje (€1,75): 6

De bestelling :
2 x aardbeientaartje
4 x miserable
6 x hoorntje
kost €24,00
```

OF

```
Assortiment gebakjes: [aardbeientaartje (€2,75), eclair (€1,50), miserable (€2,00), hoorntje (€1,75)]

Geef aantal voor aardbeientaartje (€2,75): 0
Geef aantal voor eclair (€1,50): 0
Geef aantal voor miserable (€2,00): 0
Geef aantal voor hoorntje (€1,75): 0

Niets besteld.
```

Applicatie 2:

Het assortiment gebakjes is hetzelfde als bij de vorige Applicatie.

Nu worden de bestellingen opgevraagd voor 3 klanten en deze aantallen worden bijgehouden in **één array bestellingen**. Bedoeling is dat de bakker achteraf nog altijd weet wat hij aan welke klant moet leveren.

Na de invoer wordt dan het overzicht gegenereerd waarbij de bakker per gebakje ziet hoeveel hij er moet van klaarmaken. Indien van een bepaalde soort niets is besteld, dan wordt dit ook niet vermeld in het overzicht.

Voorbeeld van mogelijke uitvoer:

```
Assortiment gebakjes: [aardbeientaartje (€2,75), eclair (€1,50), miserable (€2,00), hoorntje (€1,75)]

Bestelling voor klant 1:
Geef aantal voor aardbeientaartje (€2,75): 1
Geef aantal voor eclair (€1,50): 0
Geef aantal voor miserable (€2,00): 1
Geef aantal voor hoorntje (€1,75): 0

Bestelling voor klant 2:
Geef aantal voor aardbeientaartje (€2,75): 4
Geef aantal voor eclair (€1,50): 5
Geef aantal voor miserable (€2,00): 0
Geef aantal voor hoorntje (€1,75): 0

Bestelling voor klant 3:
Geef aantal voor aardbeientaartje (€2,75): 1
Geef aantal voor eclair (€1,50): 2
Geef aantal voor miserable (€2,00): 3
Geef aantal voor hoorntje (€1,75): 0

Totaal te bakken:
6 x aardbeientaartje
7 x eclair
4 x miserable
```