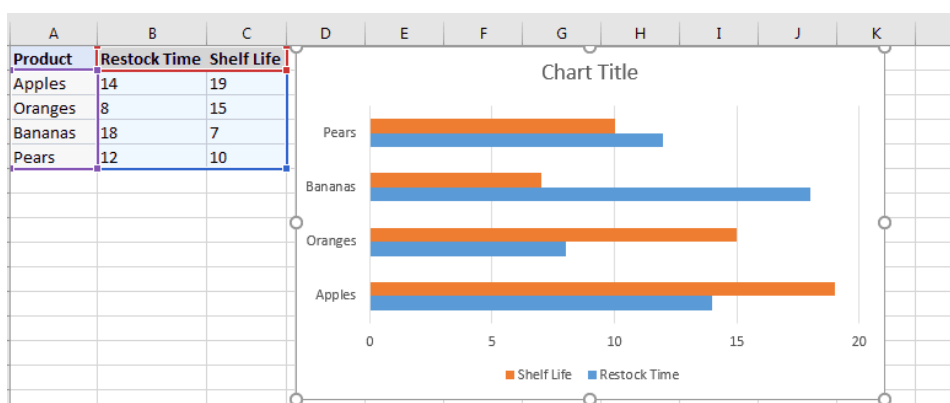


Project Java Fundamentals

Opgave : De BarChart klasse

In dit project zal je een **BarChart** klasse schrijven waarmee je een tabel bestaande uit rijen (categoriën) en kolommen (groepen) van gehele getallen kan visualiseren aan de hand van een bar chart. Het doel van een bar chart is om de frequentieverdeling van data grafisch weer te geven. Je kan horizontale (staafdiagrammen) en verticale (kolomdiagrammen) bar charts onderscheiden. Bij een kolomdiagram heb je staande balken en bij een staafdiagram liggende balken. Bij meerdere groepen van data kun je kolommen en staven naast elkaar zetten, maar ook op elkaar. In het eerste geval spreken we van een gegroepeerd kolom- of staafdiagram. In het laatste geval ontstaat een gestapeld kolom- of staafdiagram.

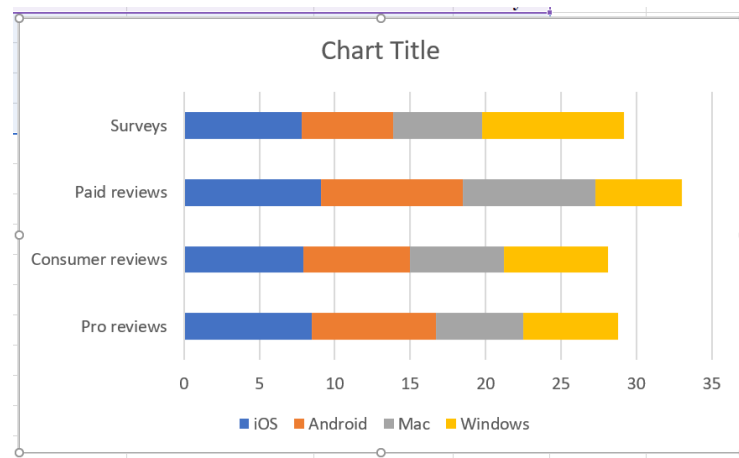
Bij een staafdiagram zijn de categorieën geplaatst op de verticale as, terwijl de schaalverdeling op de horizontale as staat. Per categorie kunnen verschillende staven weergegeven worden, zoals in onderstaande figuur 1. Dit is een voorbeeld van een gegroepeerd staafdiagram. In dit voorbeeld zijn er 2 groepen, namelijk Shelf Life en Restock Time. En er zijn 4 categorieën, namelijk Apples, Oranges, Bananas en Pears.



Figuur 1: Een gegroepeerd staafdiagram.

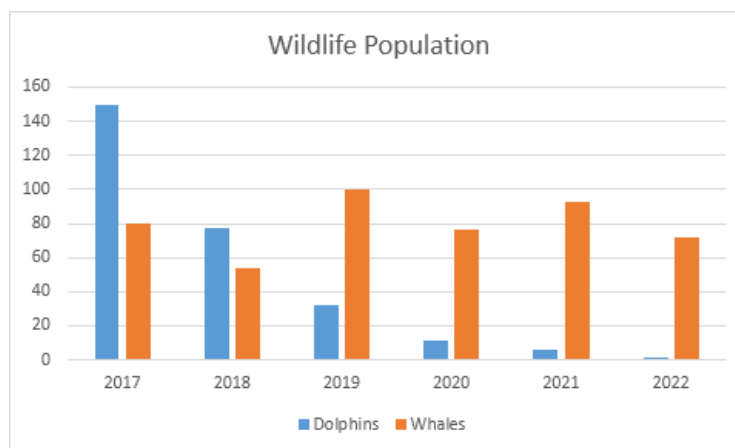
In figuur 2 zie je een voorbeeld van een gestapeld staafdiagram. In dit geval worden de verschillende staven per categorie aaneengesloten tot 1 staaf. In dit voorbeeld zijn er 4 groepen (namelijk, iOS, Android, Mac en Windows) en 4 categorieën (Surveys, Paid reviews, Consumer reviews en Pro reviews).

Een kolomdiagram heeft een verticale oriëntatie. In dit geval staan de categorieën op de horizontale as en de schaalverdeling op de verticale as. In het voorbeeld



Figuur 2: Een gestapeld staafdiagram

van figuur 3 zie je een gegroepeerd kolomdiagram. De categorieën zijn hier de jaartallen. Er zijn 2 groepen, namelijk de groep Dolphins en de groep Whales.



Figuur 3: Verticaal gegroepeerde bar chart

Scenario's

In dit project zal je code schrijven om deze charts uit te printen naar het scherm. We delen het project op in verschillende scenario's :

1. Scenario 1 : Het ingeven en weergeven van data voor 1 groep
2. Scenario 2 : Een staafdiagram tekenen
3. Scenario 3 : Een kolomdiagram tekenen

4. Scenario 4 : Het ingeven en weergeven van data voor meerdere groepen
5. Scenario 5 : Een gegroepeerd staafdiagram tekenen
6. Scenario 6 : Een gestapeld staafdiagram tekenen
7. Scenario 7 : Een gegroepeerd kolomdiagram tekenen

In de rest van het document overlopen we deze scenario's.

Verplichte datavelden en methodedefinities van je klasse BarChart

Volgende velden en methodedefinities worden verwacht in je code. Wijzig hieraan niets of je code zal **NIET** (automatisch) verbeterd kunnen worden (en dus ook geen score krijgen !!). Wat wel kan is dat je extra hulpmethoden opneemt opdat je methoden zelf niet te lang zouden worden. Splits deze dan op in zinnige deelfunctionaliteiten.

Allereerst definieer je volgende **constanten** :

```
TITLE // de titel van je bar chart
NRGROUPS // het maximum aantal groepen
```

Daarnaast voorzie je nog volgende gewone **datavelden**. Merk op: Voeg **GEEN** extra datavelden toe !

```
categories // een array van Strings
data // de data van alle groepen wordt bijgehouden in
      // een 2dim array van integer getallen
groups // de namen van de groepen worden bijgehouden
        //in een array van Strings
symbols // de symbolen waarmee een groep wordt aangeduid
        //worden bijgehouden in een array van chars
```

Voorzie vervolgens volgende **methoden** in je klasse BarChart. De eerste 2 zijn de **constructormethoden** :

```
public BarChart(String title, String[] categories,
                String groupName, char symbol, int[] data)

public BarChart(String title, int nrGroups, String[] categories)

public boolean putGroupData(String groupName, char symbol,
                           int[] data)
```

Voorzie in deze klasse 2 verschillende constructoren met parameters. De eerste gaat er vanuit dat de data slechts 1 groep bevat. In dat geval kan de data van die groep gewoon als een array van 1 dimensie meegegeven worden. Ook de naam van die groep en het symbool dat in de chart gebruikt zal worden om die groep aan te duiden worden hier meegegeven als parameter. Ook de titel die je aan de chart wil geven en de categorieën worden meegegeven.

In de tweede versie van de constructor worden naast de titel en de categorieën alleen het maximum aantal groepen meegegeven als parameter. Via deze constructor kan je objecten maken om bar charts met meerdere groepen voor te stellen. Een groep met z'n bijbehorende data zal telkens afzonderlijk moeten ingegeven worden via de methode `putGroupData`. Via deze weg kan je de data van een groep samen met zijn naam en zijn symbool toevoegen aan de juiste datavelden. **Let op, je kan geen 2 groepen met dezelfde naam of met hetzelfde symbool ingeven. Wanneer het maximum aantal groepen al ingegeven is, kan je geen extra groep ingeven.**

```
public String showData()
public String showLegend()
```

De methode `showData` geeft de titel en de volledige datatabel terug in een `String`. De `showLegend` methode maakt een legende voor de grafiek, m.a.w. voor elk symbool wordt weergegeven voor welke groep dit symbool staat. Deze 2 methoden geven telkens een opgebouwde `string` terug, zodat deze in het demo-programma uitgeprint kan worden naar het scherm.

```
private int maxValueTable(int[][] tabel)
```

De methode `maxValueTable` is een hulpmethode om in een tabel met integer getallen de grootste waarde te zoeken en deze terug te geven als resultaat. Vermoedelijk heb je deze methode nodig om een kolomdiagram tekenen.

```
public String makeChart(char orientation, boolean stacked)
```

De `makeChart` methode bouwt afhankelijk van de meegegeven parameters een chart op als een `String`. Wanneer als orientatie een 'H' wordt meegegeven zal een staafdiagram het resultaat zijn. Indien de meegegeven boolean `false` is, dan wordt in het geval van meerdere groepen een gegroepeerd staafdiagram aangemaakt. Is de meegegeven boolean `true` wordt een gestapeld staafdiagram gemaakt, uiteraard enkele bij meerdere groepen. De waarde van de boolean doet er dus niet toe indien de data slechts uit 1 groep bestaat.

Bij een verticale orientatie (de char 'V' werd meegegeven) wordt een kolomdiagram gemaakt. Hier hoeft je enkel de optie te implementeren met een boolean waarde `stacked = false` (de gegroepeerde versie dus). De gestapelde versie is buiten de scope van dit project. Merk op, resultaten worden telkens opgebouwd in een `String`, die vanuit het demo-programma geprint kunnen worden naar het scherm.

Zoals de scenario's aangeven zal je eerst charts moeten implementeren voor data bestaande uit 1 groep. In dit geval zou je gerust het interne dataveld `data` kunnen vervangen door een 1 dimensionele array van getallen. Bij uitbreiding naar meerdere groepen zal je echter meerdere dimensies nodig hebben en zal je alsnog de uitbreiding moeten maken. Om eenvoudig te starten zou je dus met een 1 dimensionele array kunnen beginnen.

Demo van de BarChart klasse

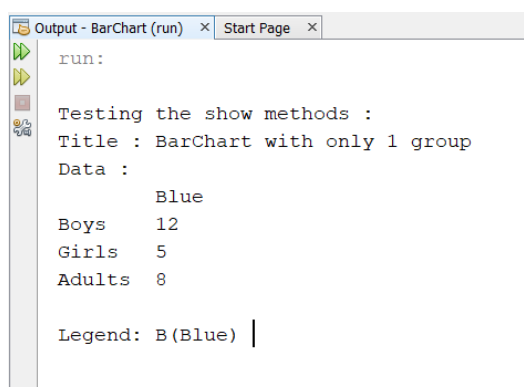
In dit onderdeel overlopen we per scenario de codefragmenten die je terugvindt in het codeblok van de main methode van de DemoBarChart klasse. Hierin kan je zien hoe de constructoren en methoden van de BarChart klasse gebruikt worden. Wanneer je de klasse BarChart implementeert is het de bedoeling dat je programma dezelfde output genereert dan degene die we in de volgende subsecties bespreken.

Scenario 1 : Het ingeven en weergeven van data voor 1 groep

```
// Scenario 1 : 1 groep
//aanmaken object + ingeven van de data

BarChart chart1group = new BarChart("BarChart with only 1 \
    →group", new String[]{"Boys", "Girls", "Adults"},
        "Blue", 'B', new int[]{12, 5, 8});

System.out.println("\nTesting the show methods : ");
System.out.println(chart1group.showData());
System.out.println(chart1group.showLegend());
```



```
Output - BarChart (run) x Start Page x
run:
Testing the show methods :
Title : BarChart with only 1 group
Data :
      Blue
Boys    12
Girls   5
Adults  8

Legend: B(Blue) |
```

Figuur 4: Printout van scenario 1

In scenario 1 wordt allereerst de constructor aangeroepen om een BarChart object aan te maken voor data die slechts bestaat uit 1 groep. Die groep krijgt de naam Blue. De data zal weergeven hoeveel personen uit de categorie van respectievelijk volwassenen (Adults), jongens (Boys) en meisjes (Girls) aangeven dat blauw hun lievelingskleur is. Op voorhand wordt de array cat1group met de categorieën klaargezet en de array data1group met de eigenlijke data. Hieruit kan je zien dat er 12 jongens aangeven dat blauw hun lievelingskleur is, 5 meisjes en 8 volwassenen.

Vervolgens wordt deze info uitgeprint via de showData methode. Als laatste wordt de showLegend methode getest. Deze zal de legende van de grafiek voorstellen en schrijft na het symbool dat gekozen werd voor deze specifieke groep, de naam van de groep voluit : B(Blue). De printout van de code onder scenario 1 vind je terug in Figuur 4. Dit

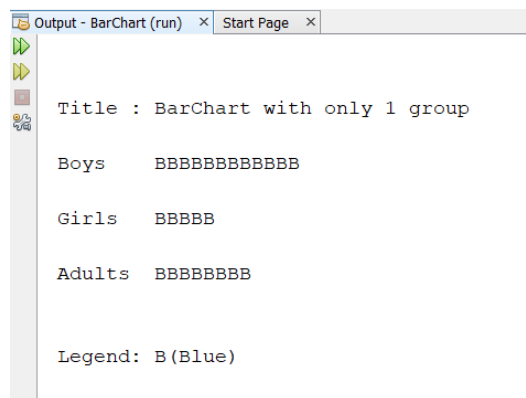
is exact wat jouw programma ook moet uitprinten bij runnen van dit codefragment uit de `DemoBarChart` klasse.

Scenario 2 : Een staafdiagram tekenen

```
// Scenario 2 : staafdiagram 1 groep

System.out.println(chart1group.makeChart('H', false));
```

In scenario 2 gebeurt een oproep van de methode `makeChart` methode naar het object `chart1group` om een staafdiagram tekenen met de data die in vorig scenario ingegeven werd. Dit kan door de char 'H' als argument mee te geven en geeft de tekening die je terugvindt in Figuur 5.



Figuur 5: Printout van scenario 2 - staafdiagram op basis van de data uit Figuur 4

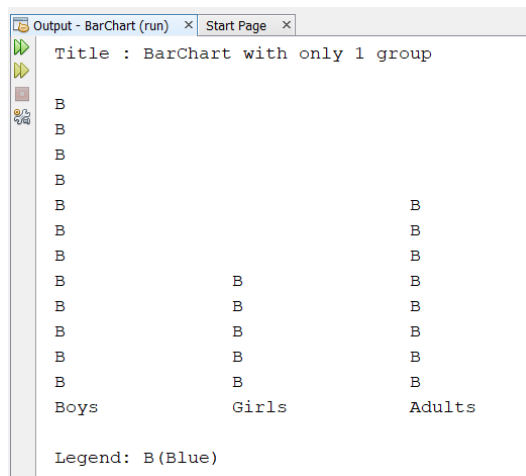
Merk op dat de grafiek ook de titel en legende omvat. De `makeChart` methode doet dus een oproep naar de methode `showLegend`.

Scenario 3 : Een kolomdiagram tekenen

```
//Scenario 3 : kolomdiagram 1 groep

System.out.println(chart1group.makeChart('V', false));
```

Analoog aan vorig scenario kan nu een kolomdiagram getekend worden door de methode `makeChart` op te roepen op het object `chart1group` met als oriëntatie de char 'V'. Het resultaat hiervan is terug te vinden in Figuur 6. Ook hier wordt de titel getoond en onderaan de legende toegevoegd.



Figuur 6: Printout van scenario 3 - een kolomdiagram op basis van de data uit Figuur 4

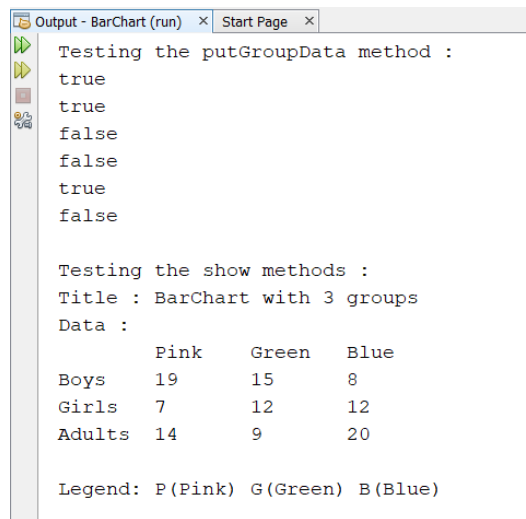
Scenario 4 : Het ingeven en weergeven van data voor meerdere groepen

```
//Scenario 4 : meerdere groepen
//aanmaken object
BarChart chart3groups = new BarChart("BarChart with 3 \
    →groups", 3, new String[]{"Boys", "Girls", "Adults"});

// ingeven van data per groep
System.out.println("\nTesting the putGroupData method : ");
System.out.println(chart3groups.putGroupData("Pink", 'P', \
    →new int[]{19, 7, 14})); //ok
System.out.println(chart3groups.putGroupData("Green", 'G', \
    →new int[]{15, 12, 9})); //ok
System.out.println(chart3groups.putGroupData("Green", 'Y', \
    →new int[]{15, 12, 9})); //nok groep bestaat al
System.out.println(chart3groups.putGroupData("Yellow", 'G', \
    →new int[]{15, 12, 9})); //nok symbool bestaat al
System.out.println(chart3groups.putGroupData("Blue", 'B', \
    →new int[]{8, 12, 20})); //ok
System.out.println(chart3groups.putGroupData("Orange", 'O', \
    →new int[]{15, 12, 9})); //nok max 3 groepen

System.out.println("\nTesting the show methods : ");
System.out.println(chart3groups.showData());
System.out.println(chart3groups.showLegend());
```

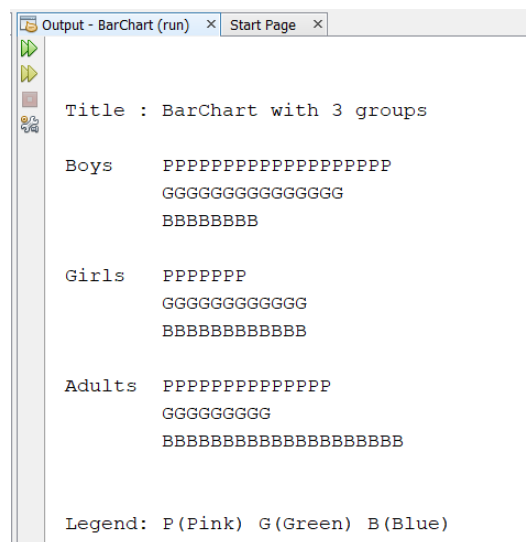
In scenario 4 toont de code in de DemoBarChart klasse hoe je de tweede constructor moet gebruiken wanneer je data bestaande uit meerdere groepen wil ingeven. Via de constructor geef je de titel mee, het maximum aantal groepen en de array met categorieën. De data zelf wordt per groep apart ingegeven via de methode putGroupData. Deze moet voldoen aan enkele voorwaarden, vandaar dat niet alle oproepen toegestaan zijn. De output kan je vinden in Figuur 7.



Figuur 7: Printout van scenario 4

Scenario 5 : Een gegroepeerd staafdiagram tekenen

```
// Scenario 5 gegroepeerd staafdiagram
System.out.println(chart3groups.makeChart('H', false));
```



Figuur 8: Printout van scenario 5 : een gegroepeerd staafdiagram

Opnieuw moet een oproep naar de methode `makeChart` methode met parameters 'H' en `false` volstaan om een staafdiagram te tekenen. Vermits het dataobject (`chart3groups`) uit meerdere groepen van data bestaat, zal nu automatisch een gegroepeerde versie getekend worden, zie Figuur 8. Opnieuw wordt de titel en een legende mee uitgeprint.

Quotering

Deel 1 : functionele werking + aanpak code (/ 18)

- Scenario 1 : /2
 1. Constructor methode voor 1 groep (/1)
 2. uitprinten van de data (/0.5)
 3. uitprinten van de legende (/0.5)
- Scenario 2 : / 2
 - weergeven en implementatie van een staafdiagram (1 groep)
- Scenario 3 : / 2
 - weergeven implementatie van een kolomdiagram (1 groep)
- Scenario 4 : /4
 1. Constructor methode voor meerdere groepen (/1)
 2. ingeven data (via putGroupData methode) (/2)
 3. uitprinten van de data (/ 0.5)
 4. uitprinten van de legende (/0.5)
- Scenario 5 : /3
 - weergeven en implementatie van een gegroepeerd staafdiagram
- Scenario 6 : /3
 - weergeven en implementatie van een gestapeld staafdiagram
- Scenario 7 : /2
 - weergeven en implementatie van een gegroepeerd kolomdiagram

Deel 2 : Documentatie (/2)

Gebruik javadoc om je klasse te documenteren. Genereer deze documentatie in html.

Opgelet! Principiële fouten tegen een goede programmeerstijl zullen ook in rekening gebracht worden (-0.5 punt per soort overtreding; maximaal - 4). Let daarom zeker op volgende aspecten:

- indentatie en spaciëring
- gebruik van variabelen (naamgeving en gepast type)
- leesbare code, gemakkelijk aanpasbaar voor anderen

- methoden met een duidelijke functionaliteit : maak je methoden niet te lang, splits deelfunctionaliteiten af in afzonderlijke methoden

Richtlijnen en afspraken

- Op Toledo vind je een gezipte versie van een Netbeansproject waarvan je mag vertrekken. Hierin vind je alvast de klasse DemoBarChart en een lege klasse BarChart die je zal moeten vervullen.
- Voorzie javadoc documentatie bovenaan je klasse, alsook boven elk veld en elke methode.
- Genereer de html-pagina's met behulp van javadoc. Zorg ervoor dat de volledige html documentatie onder de folder `dist` van je Netbeansproject te vinden is.
- Zorg ervoor dat je programma compileert.
- Plaats je afgewerkte netbeansproject met bijhorende html pagina's in een zip archief met volgende naamconventie: **FamilienaamVoornaam.zip**. Je project moet ten laatste tegen **Maandag 7 januari 2019 20:00u** geüpload zijn naar Toledo.
- Bij het betrappen op fraude en/of kopiëren zal er een fraudedossier worden opgesteld en voorgelegd worden aan de examencommissie.

Veel codeerplezier en succes !!