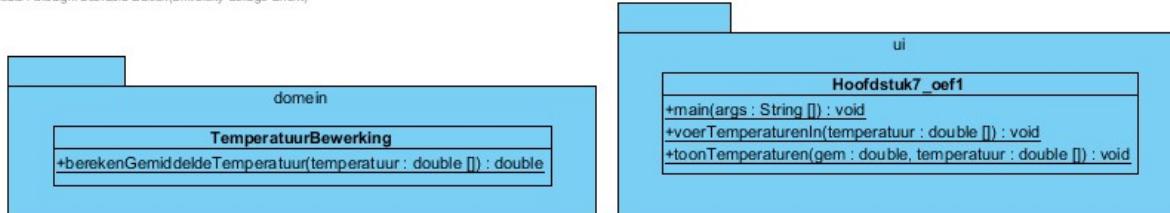


Oefening 1. Maak een applicatie die het volgende doet:

Geef 7 temperaturen in. De gemiddelde temperatuur wordt weergegeven. Vervolgens wordt per temperatuur weergegeven hoeveel graden ze onder of boven de gemiddelde temperatuur liggen. De temperaturen worden steeds met één cijfer na de komma weergegeven.

Visual Paradigm Standard Edition(University College Ghent)



Klasse TemperatuurBewerking (domein):

- methode berekenGemiddeldeTemperatuur: geeft de gemiddelde temperatuur terug. Gebruik een enhanced for-statement.

Klasse Hoofdstuk7_oef1:

De array wordt als lokale variabele gedefinieerd in de methode “main” Schrijf de methoden `voerTemperaturenIn` en `toonTemperaturen`.

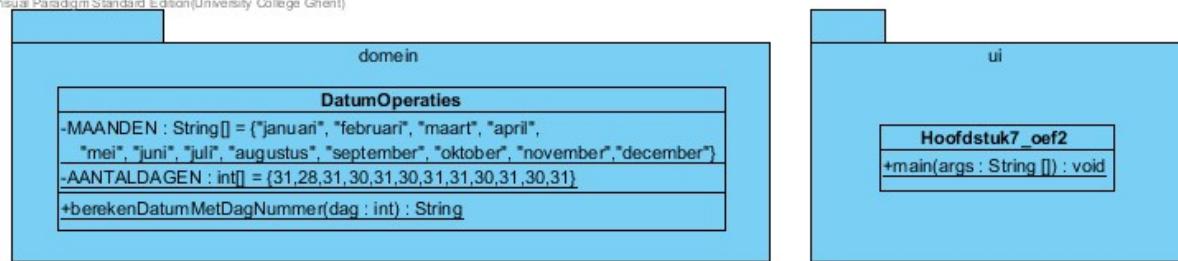
- methode voerTemperaturenIn: de temperaturen worden ingegeven.
- methode toonTemperaturen: geeft de gemiddelde temperatuur weer op scherm. Tevens geeft de methode alle temperaturen en hoeveel graden ze onder of boven het gemiddelde liggen, weer. Zie schermafdruk hieronder!

The screenshot shows the output of the application. It starts with the command `run:`. Then it prompts for 7 temperatures: 15, 16,2, 14,8, 5,9, 11,3, 16,7, and 19. The average temperature is then printed as 14,1. Below this, a table is displayed showing each temperature, its deviation from the average (graden), and whether it is above or below the average (verschil). The table is as follows:

temperatuur	graden	verschil
1	15,0	0,9 graden BOVEN het gemiddelde
2	16,2	2,1 graden BOVEN het gemiddelde
3	14,8	0,7 graden BOVEN het gemiddelde
4	5,9	8,2 graden ONDER het gemiddelde
5	11,3	2,8 graden ONDER het gemiddelde
6	16,7	2,6 graden BOVEN het gemiddelde
7	19,0	4,9 graden BOVEN het gemiddelde

BUILD SUCCESSFUL (total time: 22 seconds)

Oefening 2. DatumApplicatie



Een getal tussen 1 en 365 wordt aan de gebruiker gevraagd (invoercontrole voorzien). Het getal stelt het dagnummer van het jaar (geen schrikkeljaar!) voor. De overeenkomstige datum wordt weergegeven.

Voorbeelden:

```

: Output - Hoofdstuk7_oef2 (run)
|> run:
|> Geef dagnummer van het jaar in: 90
|> dag 90 = 31 maart
|> BUILD SUCCESSFUL (total time: 4 seconds)

: Output - Hoofdstuk7_oef2 (run)
|> run:
|> Geef dagnummer van het jaar in: 400
|> Geef dagnummer van het jaar in: 300
|> dag 300 = 27 oktober
|> BUILD SUCCESSFUL (total time: 9 seconds)

```

The code block shows two terminal windows for the 'Hoofdstuk7_oef2' application. The first window shows the user inputting '90' and receiving the output 'dag 90 = 31 maart'. The second window shows the user inputting '400' and '300', with the output 'dag 300 = 27 oktober'. Both windows end with a 'BUILD SUCCESSFUL' message.

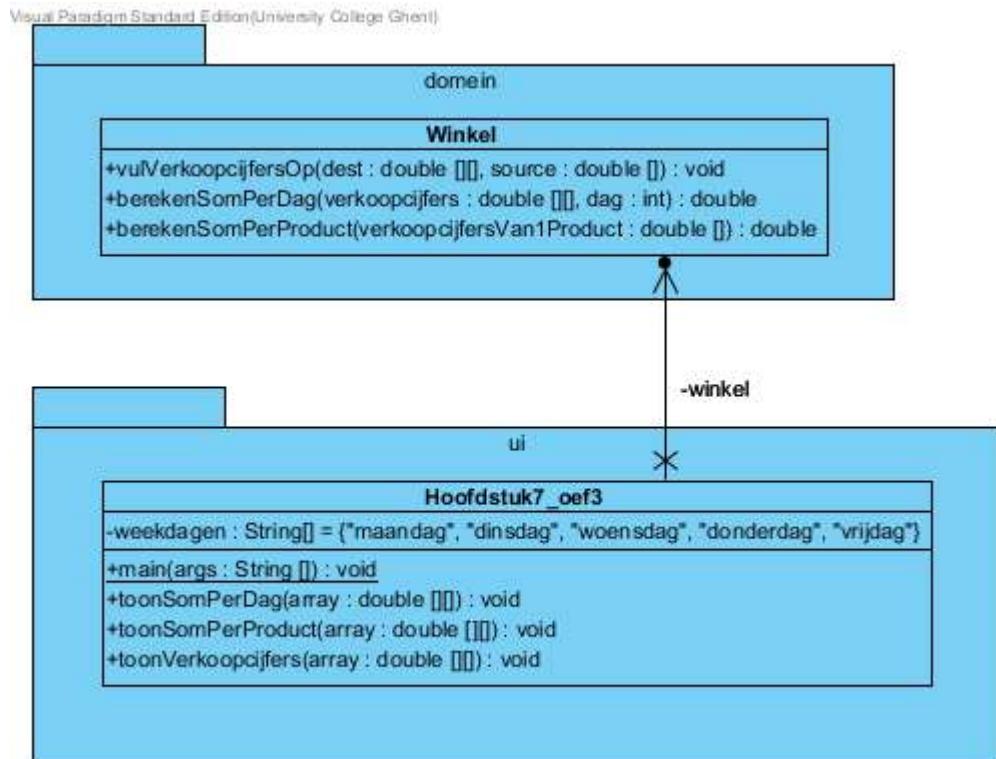
Schrijf de domeinklasse “**DatumOperaties**”:

- **Twee attributen:** de *constante* arrays
 - MAANDEN: {"januari", "februari", "maart", "april", "mei", "juni", "juli", "augustus", "september", "oktober", "november", "december"}
 - AANTAL_DAGEN: {31,28,31,30,31,30,31,31,30,31,30,31}
- Eén *static* methode: **berekenDatumMetDagNummer**: de methode zet een getal om in een datum. Bvb. het getal 105 wordt omgezet naar “15 april”.

Schrijf de applicatie “**Hoofdstuk7_oef2**”: zorgt voor de in- en uitvoer

Oefening 3. WinkelApplicatie

In een kleine winkel worden de verkoopcijfers van 3 producten, per weekdag bijgehouden. De som van de verkoopcijfers worden per dag en per product weergegeven. Tevens worden alle verkoopcijfers weergegeven (zie schermafdruk).



Schrijf de klasse “Winkel” en “Hoofdstuk7_oef3”.

Winkel

Schrijf de non-static methoden **vulVerkoopcijfersOp**, **berekenSomPerDag** en **berekenSomPerProduct**.

- **Methode vulVerkoopcijfersOp:** de waarden van de array “cijfers” worden in de array van “verkoopcijfers” als volgt geplaatst:
verkoopcijfers[0][0] = cijfers[0], verkoopcijfers[0][1] = cijfers[1],
verkoopcijfers[0][2] = cijfers[2], verkoopcijfers[0][3] = cijfers[3],
verkoopcijfers[0][4] = cijfers[4], verkoopcijfers[1][0] = cijfers[5],
verkoopcijfers[1][1] = cijfers[6], verkoopcijfers[1][2] = cijfers[7], etc.
- **Methode berekenSomPerDag:** geeft de som van de verkoopcijfers van één dag terug (som van de cijfers in een kolom).
- **Methode berekenSomPerProduct:** geeft de som van de verkoopcijfers van één product terug (som van de cijfers van de volledige array, die een rij voorstelt).

Hoofdstuk7_oef3

De klasse bevat twee attributen: de constante één-dimensionale array “**weekdagen**”. Deze wordt geïnitialiseerd op {"maandag", "dinsdag", "woensdag", "donderdag", "vrijdag"}. Daarnaast heb je ook nog het (static) attribuut winkel dat voorzien wordt.

De twee-dimensionale array “**verkoopcijfers**” wordt als **lokale variabele** gedefinieerd in de methode “main”. In deze methode wordt ook de één-dimensionale array “**cijfers**” geïnitialiseerd op {100.50, 500, 700, 0, 333.95, 1000, 489, 789.10, 90, 50000, 66, 0, 700, 20, 30}; Het attribuut winkel wordt hier geïnstantieerd.

Schrijf de non-static methoden **toonSomPerDag**, **toonSomPerProduct**, **toonVerkoopcijfer**.

- **methode toonSomPerDag**: geeft de som van de verkoopcijfers per dag op het scherm weer. Maak ook gebruik van de array “weekdagen”. Per dag wordt de methode “**berekenSomPerDag**” opgeroepen.
- **methode toonSomPerProduct**: geeft de som van de verkoopcijfers per product op het scherm weer. Per product wordt de methode “**berekenSomPerProduct**” opgeroepen. Het argument is een **rij van de tabel**.
- **methode toonVerkoopcijfers**: de verkoopcijfers van alle producten per dag worden weergegeven. Maak ook gebruik van de array “weekdagen”.

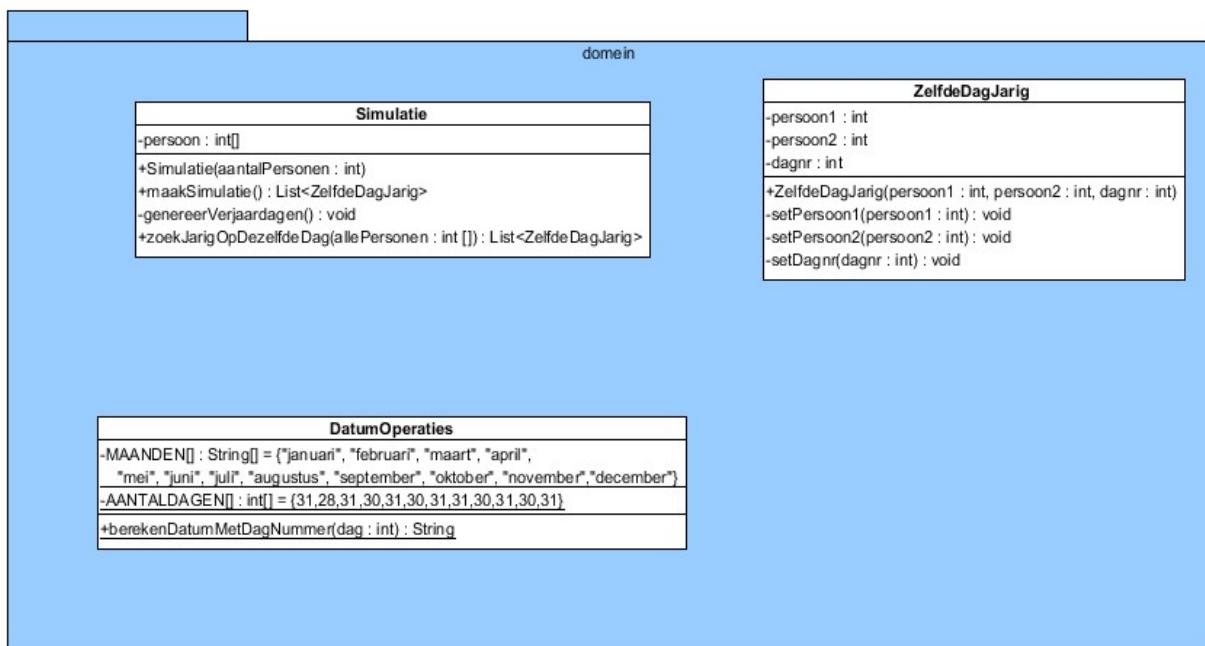
```
: Output - Hoofdstuk7_oef3 (run)
run:
TOTAAL
maandag : 1166,50
dinsdag : 989,00
woensdag : 2189,10
donderdag : 110,00
vrijdag : 50363,95

TOTAAL
product 1 : 1634,45
product 2 : 52368,10
product 3 : 816,00

      maandag   dinsdag   woensdag   donderdag   vrijdag
product 1    100,50     500,00     700,00      0,00    333,95
product 2   1000,00     489,00     789,10     90,00  50000,00
product 3     66,00      0,00     700,00     20,00     30,00
BUILD SUCCESSFUL (total time: 0 seconds)
```

Oefening 4: Jarigen zoeken

Maak het simulatieprogramma ‘jarigen zoeken’. Alle personen, die op dezelfde dag jarig zijn, worden weergegeven.



Schrijf de domeinklasse “ZelfdeDagJarig”

- 3 attributen: persoon1 (nummer van de persoon), persoon2 (nummer van de persoon), dagnr (nummer van de dag van het jaar waarop ze jarig zijn, we werken zonder schrikkeljaren). Zorg voor controle bij dagnr!
- Alle getters en setters voor deze attributen

Schrijf de domeinklasse “Simulatie”.

- **Eén attribuut** “personen” = een array van type int. Voor elke persoon zal zijn verjaardag als dagnummer worden bijgehouden.
- **Constructor met één argument:** aantal personen van een groep. De array (=attribuut) wordt gecreëerd.
- **public List<ZelfdeDagJarig> maakSimulatie():** de methode roept “genereerVerjaardagen” en “zoekJarigenOpDezelfdeDag” op. De methode geeft de lijst van ZelfdeDagJarig terug.
- **private methode genereerVerjaardagen:** genereert voor elke persoon een getal tussen 1 en 365 (het attribuut wordt opgevuld).
- **public List<ZelfdeDagJarig> zoekJarigenOpDezelfdeDag (int[] allePersonen):** de methode zoekt alle personen die op dezelfde dag jarig zijn. De lijst waarin gezocht wordt, geef je als parameter door. Het resultaat wordt in een ArrayList bewaard en teruggegeven.
Zie werkwijze volgende pagina

Werkwijze:

Stel dat het attribuut “persoon” opgevuld is met {10, 60, 10, 34, 60}

Creëer een lege arrayList.

Vergelijk **persoon1**

met persoon2: 10 != 60 ➔ er gebeurt niks

met persoon3: **10 == 10 ➔**

voeg aan de arrayList een object van de klasse ZelfdeDagJarig toe

met attributen persoon1 = 1, persoon2 = 3 en dagnr = 10

Vergelijk **persoon2**

met persoon3: 60 != 10 ➔ er gebeurt niks

met persoon4: 60 != 34 ➔ ook niks

met persoon5: **60 == 60 ➔**

voeg aan de arrayList een object van de klasse ZelfdeDagJarig toe

met attributen persoon1 = 2, persoon2 = 5 en dagnr = 60

Vergelijk **persoon3**

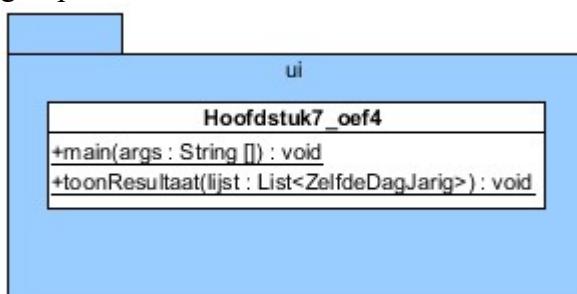
met persoon4: 10 != 34 ➔ er gebeurt niks

met persoon5: 10 != 60 ➔ er gebeurt niks

Vergelijk **persoon4**

met persoon5: 34 != 60 ➔ er gebeurt niks

Schrijf de applicatie “**Hoofdstuk7_oef4**”: het simulatieprogramma wordt uitgetest voor een groep van **60** mensen.



Het resultaat (arrayList) wordt weergegeven op het scherm (zie schermafdruk). Gebruik hiervoor de methode **toonResultaat**. Deze methode maakt ook gebruik van de domeinklasse **DatumOperaties** (zie oefening 2).

Mogelijke output:

```
Output - Hoofdstuk7_oef4 (run) %
run:
      persoon   persoon     verjaardag
          9       25     21 december
          14      38      1 oktober
          26      55     18 september
          54      59     29 september
BUILD SUCCESSFUL (total time: 0 seconds)
```

Of wanneer er meer dan 2 jarigen op dezelfde dag zijn:

```
Output - Hoofdstuk7_oef4 (run) ✘

run:
    persoon    persoon    verjaardag
        6          57      31 mei
        11         39      22 juli
        19         23       6 juli
        24         32      16 maart
        32         56      16 maart
BUILD SUCCESSFUL (total time: 0 seconds)
```