

Demografie

We wensen informatie bij te houden over het aantal inwoners van een aantal steden van een regio.

Maak een klasse met de naam **Stad**. Een object van deze klasse heeft als eigenschappen de naam van een stad en het aantal inwoners. Het aantal inwoners is een positief geheel getal en moet minimaal 70 zijn.

Je moet een object van deze klasse kunnen maken op basis van beide eigenschappen of enkel op basis van de naam van de stad, het aantal inwoners is dan 5000.

Twee Stad objecten zijn gelijk als ze dezelfde naam hebben.

Voorzie een functionaliteit met de naam *heeft_meer_inwoners* om na te gaan of een Stad-object meer inwoners heeft dan een ander Stad-object.

Voorzie een functionaliteit om de eigenschappen van een Stad object in string vorm weer te geven naar volgend voorbeeld:

"Leuven heeft 102000 inwoners"

Maak een klasse met de naam **Demografie** die de naam van een regio bevat en een lijst bevat van Stad-objecten.

Initieel is deze lijst leeg. De constructor krijgt enkel de naam van de regio mee.

Je moet een functionaliteit met de naam *voeg_stad_toe* voorzien om een nieuw Stad object aan de lijst toe te voegen. Het nieuwe Stad-object wordt zodanig toegevoegd dat de lijst gesorteerd is op aantal inwoners. Je mag geen Stad object toevoegen met dezelfde naam als een reeds bestaand Stad object in de lijst.

Je moet een functionaliteit met de naam *bereken_totaal_aantal_inwoners* voorzien om het totaal aantal inwoners te berekenen over alle Stad-objecten van de lijst.

2 steden gaan fusioneren. Je moet een functionaliteit met de naam *fusioneer* voorzien om op basis van de naam van deze twee steden en een nieuwe naam de informatie in de lijst aan te passen.

Voeg een functionaliteit met de naam `geef_steden_met_minimum_aantal_inwoners` toe die een lijst teruggeeft van alle steden met een gegeven aantal minimum inwoners.

Je moet een functionaliteit voorzien om een string vorm weer te geven van een Demografie object als volgt:

eerste lijn : "Regio: Vlaanderen", tweede lijn: "Totaal aantal inwoners: 524555", volgende lijnen: per lijn een string vorm van een Stad-object uit de lijst van Stad-objecten volgens het voorbeeld in de vorige paragraaf.

Schrijf volgend stukje code in een apart bestand met de naam `app.py`:

- maak een Stad object met als stadsnaam "Leuven" en aantal inwoners 102000
- maak een Stad object met als naam "Duh" met 5000 inwoners
- maak een Stad object met als stadsnaam "Gent" en aantal inwoners 263429
- maak een Stad object met als stadsnaam "Hasselt" en aantal inwoners 80000
- maak een Stad object met als stadsnaam "Genk" en aantal inwoners 85000
- maak een Stad object met als stadsnaam "Genk" en aantal inwoners 5000
- maak een Demografie object met als regio naam "Vlaanderen" en voeg de stads objecten uit vorige puntjes in die volgorde toe
- schrijf de string vorm van dit Demografie object uit

Gewenste uitvoer:

```
Regio: Vlaanderen
Totaal aantal inwoners: 535429
Duh heeft 5000 inwoners
Hasselt heeft 80000 inwoners
Genk heeft 85000 inwoners
Leuven heeft 102000 inwoners
Gent heeft 263429 inwoners
```

- de stad Gent en de stad Duh gaan fusioneer tot de stad GentPlus, pas de lijst aan
- schrijf de string vorm van dit Demografie object uit

Gewenste uitvoer:

Regio: Vlaanderen
Totaal aantal inwoners: 535429
Hasselt heeft 80000 inwoners
Genk heeft 85000 inwoners
Leuven heeft 102000 inwoners
GentPlus heeft 268429 inwoners

- print een lijst van steden met minimaal 100000 inwoners

Gewenste uitvoer:

steden met meer dan 100000 inwoners:
Leuven heeft 102000 inwoners
GentPlus heeft 268429 inwoners