

## Schrijf je code in het Engels.

### Gebruik naming conventions!

### Labo week 3

Lesdoelen:

- Werken met console
- Collections gebruiken

### **Maak een nieuw project aan per labo!**

### **Oefening 1 : AnimalShelter**

Herneem de code van je dierenasiel (AnimalShelter) van vorig labo.

Schrijf een console toepassing om het asiel te beheren.

Bij opstart van je toepassing toon je een welkombericht en nadien een menu met volgende opties:

- 1 : voeg een kat toe
- 2 : voeg een hond toe
- 3 : toon alle dieren
- 4 : toon dier
- 5 : verwijder dier
- 6 : einde

Zorg ervoor dat je na het kiezen en uitvoeren een menuoptie telkens opnieuw de menu te zien krijgt. Enkel als je '6' intikt, stopt je programma. Denk aan een while lus om dit op te lossen.

Bij opties 1 en 2 vraag je aan de gebruiker de gegevens van de kat of hond. Nadien maak je de kat of hond aan en voeg je deze toe aan het asiel.

Optie 3 voert de code uit om alle dieren te tonen.

Bij opties 4 en 5 vraag je een id aan de gebruiker en dan voer je respectievelijk de show en delete code uit.

Uitbreiding: zorg ervoor dat je geen duplicaten kunt toevoegen. Controleer telkens je een kat of hond toevoegt of er al een dier met die gegevens in zit. Als je een duplicaat tegenkomt, waarschuw dan de gebruiker.

In welke class of classes heb je een equals methode gezet?

## Oefening 2 : Afspraken

De dierenarts wil een elektronisch systeem voor zijn afspraken.

Maak een nieuwe package aan voor deze oefening.

Maak een nieuwe class afspraak (**appointment**) aan. Een afspraak heeft als verplichte attributen naam klant (**customer name**) en reden (**reason**).

Maak een nieuw class agenda (**agenda**) aan en voeg een method toe om een afspraak aan de agenda toe te voegen.

De dierenarts wenst de afspraken in volgorde van toevoegen af te werken.

Voeg de juiste collection toe aan je agenda class en voorzie een method op je agenda class om die afspraak te tonen en uit de lijst van afspraken te halen.

Als er geen afspraken meer zijn, toon je "Er zijn geen afspraken".

Soms gebeurt het dat er een dringende afspraak tussen komt. Voorzie op je agenda een method om een afspraak vooraan toe te voegen.

Test dit in je main class.

## Oefening 3 : Afspraken per dag

Momenteel kan onze dierenarts maar voor één dag zijn afspraken inplannen. Hij had graag zijn afspraken voor een latere datum op voorhand ingepland, gegroepeerd per dag.

Maak een nieuwe class week agenda aan en zorg voor een methode om een werkdag toe te voegen. Deze methode neemt een datum als input om de werkdag te identificeren. Zorg ervoor dat de dierenarts dit meermaals kan doen.

Voorzie in de week agenda een data structuur om per werkdag een agenda object te onderhouden.

Schrijf dan een methode om voor een bepaalde dag een afspraak toe te voegen en eveneens een methode om voor een bepaalde dag de eerstvolgende afspraak te tonen.

Test dit in je main class.

## Oefening 4 : Het jarige nijlpaard

Nijlpaard Hermien is jarig ( [https://www.gva.be/cnt/dmf20220926\\_95385047](https://www.gva.be/cnt/dmf20220926_95385047) )

Maak een nieuwe package aan om het verjaardagsfeestje te plannen en voorzie bij je dieren een nieuwe diersoort nijlpaard (**hippo**).

Maak een nieuwe class aan voor het feestje (**party**).

Nodig alle dieren classes uit de Animals package uit (Cat, Dog, Lion, Monkey en uiteraard Hippo) voor het feestje. Voorzie een collection (kies welke het beste past) in je feestje class om de genodigden bij te houden en een methode om de genodigden toe te voegen aan het feestje.

Eerst wordt er taart gegeten. Schrijf hier een method voor.

Er zijn 8 stukken taart. De dieren mogen om de beurt een stuk nemen. De volgorde is : eerst

de jarige Hermien, uiteraard, en dan van oud naar jong. Als iedereen een stuk heeft gehad, mag er een tweede stuk volgens de juiste volgorde gegeten worden. Elke keer dat een dier een stuk taart eet, toon je in de console: '<naam> eet een stuk taart.'. Niet iedereen zal een tweede stuk taart krijgen, toon ook in je console als een dier geen stuk krijgt : '<naam> krijgt geen taart'.

Pas je methode aan zodat het aantal stukken taart een input parameter wordt en test nu je code met bv 12 stukken taart.

Nadien zingen alle dieren een liedje voor Hermien. Ze zingen 'happy birthday to you – happy birthday to you'. Uiteraard kunnen dieren niet echt zingen, dus op elke lettergreep maakt een dier zijn geluid. De lettergrepen zijn 'ha-ppy-birth-day-to-you-ha-ppy-birth-day-to-you'. Hermien zingt niet mee, enkel de kat, hond, leeuw en aap zingen. De dieren zingen niet in vaste volgorde maar elk dier moet een geluid gemaakt hebben voor eenzelfde dier een volgende keer zijn geluid mag maken. Kies dus random een dier uit de lijst en doe dat tot elke dier zijn geluid heeft gemaakt en begin dan opnieuw.

Test je code in een main class.