

## Schrijf je code in het Engels.

### Gebruik naming conventions!

### Labo week 8

Lesdoelen:

- Werken met Threads
- Werken met Hibernate (herhaling)
- Werken met files (herhaling)

Maak voor elk labo een nieuw project aan!

### Oefening 1 : Bank

1. Maak een package 'bank'. In deze package maak je een nieuwe class Account, die een bankrekening voorstelt met een rekeningnummer en bedrag. Initieel staat er op de rekening 100 euro. Je hebt een method nodig om geld af te halen (withdraw) en een method om te storten (deposit). Bij deze functies is steeds een persoon gekoppeld die dan het geld van de rekening haalt of erop zet. Laat in je code telkens na elke handeling de code 1 seconde wachten mbv *thread.sleep()*. Check bij afhaling uiteraard of er voldoende geld op de rekening staat.

Test je code met deze main:

```
class Main {  
  
    public static void main(String[] args) {  
  
        // object creatie  
        Account obj = new Account();  
  
        // Transacties  
        obj.withdrawn("Sofia", 20);  
        obj.withdrawn("Jeroen", 40);  
        obj.deposit("Rupta", 35);  
        obj.withdrawn("Bruno", 80);  
        obj.withdrawn("Kesha", 40);  
    }  
}
```

2. Pas je code aan zodat de verschillende gebruikers simultaan geld kunnen afhalen en storten met behulp van Threads. Schrijf hiervoor 2 nieuwe classes: een ThreadDeposit en een ThreadWithdrawal, beide hebben als functie om een bepaald bedrag af te halen / te storten indien ze worden gerund. Pas je Main aan waar nodig om de nieuwe functionaliteiten te gebruiken. Wat is het gedrag van de output van de Main vergeleken met deel 1 van de oefening ?
3. Pas je code aan zodat er geen fouten kunnen gebeuren met het bedrag op de rekening.

## Oefening 2 : Patiënten en ziektefiches

Onze dierenarts wil een toepassing om ziektefiches voor zijn patiënten te registeren.

Maak twee nieuwe tabellen aan : patiënt (Patient) en ziektefiche (Card).

Een patiënt heeft 2 attributen : naam (name) en type (type). Type is een enumeratie CAT en DOG.

Een ziektefiche heeft 3 attributen : datum (date), reden van bezoek (reason) en behandeling (treatment).

Leg een bidirectionele relatie van patiënt naar ziektefiche, dus een patiënt heeft meerdere ziektefiches.

Maak nu een console toepassing aan.

Bij opstart lees je het bestand patienten.txt (download dit van Digitap) in en je vult hiermee de Patient tabel.

Je console toepassing heeft volgende opties:

1. Toon alle patiënten
2. Toon ziektefiches van patiënt
3. Voeg ziektefiche voor patiënt toe

Bij optie één toon je voor alle patiënten uit de database de ID, naam en type. Niet de ziektefiches.

Bij optie twee vraag je een ID van een patiënt en toon je enkel voor die patiënt de ziektefiches.

Bij optie drie vraag je volgende gegevens : ID patiënt, datum, reden van bezoek en behandeling. Zorg er dan voor dat de gegevens in de database komen.

Extra: is de gebruikte relatie de meest ideale? Zou je beter een andere relatie leggen?

## Oefening 3 : Deadlock (uitbreiding)

Download van Digitap het deadlock bestand.

Bekijk de code in Deadlock.java en voer deze uit. Wat gebeurt er ? Pas de code aan zodat ze niet meer blokkeert.