

Assignment 28 – Account

Doel

Het doel van deze opdracht is het oefenen met subclasses en superclasses, met het juiste gebruik van access modifiers.

Overzicht

Deze opdracht heeft 3 classes:

- Class Assignment28
- Class Account
- Class CheckingAccount

1. Voorbereiding

- a. Maak een package aan met de naam **assignment28** (let op packages beginnen altijd met een kleine letter)
- b. Maak in de package een nieuwe class met de naam **Assignment28** en maak de **main** methode aan. De verdere implementatie van de **main** methode wordt verder op in de opdracht besproken.

2. Class Account

We beginnen met het maken van de (super)class **Account**, omdat deze gebruikt gaat worden door de class **CheckingAccount**.

- a. Maak in de package **assignment28** een nieuwe class met de naam **Account**.
- b. De class **Account** heeft twee eigenschappen (fields), resp. **name** en **balance**. Bepaal zelf meest voor de hand liggend data type.
- c. Maak een methode ‘printOverview’ die een overzicht print van de fields **name** en **balance**.

Output (qua format):

```
Overview: name = Spaarrekening, balance = 5000
```

3. Class CheckingAccount

Vervolgens maken we de (sub)class **CheckingAccount**, die de eigenschappen en methoden erft van **Account**.

- a. Maak in de package **assignment28** een nieuwe class met de naam **CheckingAccount**. Deze class erft de eigenschappen en methoden van **Account**.

- b. De class **CheckingAccount** heeft een constructor waarbij je een input parameter kan meegeven: **initialBalance**. Deze input parameter bepaalt de eigenschap **balance**.

Verder geeft de constructor het field **name** de waarde “CheckingAccount”.

- c. Welke access modifiers kan je gebruiken in de class **Account** voor de fields **name** en **balance**?

```
{access modifier} {data type} name;  
{access modifier} {data type} balance;
```

- d. Welke access modifier van de werkende access modifiers vind jij het beste en waarom?

4. Class Assignment28

De **main** methode hebben we al aangemaakt in de Voorbereiding (1b). Nu gaan we de implementatie doen van de **main** methode.

- a. Instantieer van de class **CheckingAccount**, een nieuw object met de naam **myCheckingAccount** en geef een **initialBalance** mee van 5000 als input parameter van de constructor.
- b. Geeft het object **myCheckingAccount** ook een naam.
- c. Roep de methode **printOverview()** aan van het aangemaakte object **myCheckingAccount**. Welke output krijg je?