

Oefening 1 - Rekening

Zie Pijlers van OO - Werkcollege pdf oefening 2.2

Oefening 1 - Uitbreiding 1

Maak een klasse RekeningenRepository aan. Voorzie in deze klasse een methode GeefRekeningen() die een lijst met rekeningen terug geeft, zorg dat dit zowel spaar- als zichtrekeningen zijn. De waarden van de rekeningen mag je zelf kiezen.

Roep de GeefRekeningen() functie aan in de RekeningApplicatie en print voor elk Rekening object dat je terug krijgt de ToString() methode uit in de console.

Oefening 1 - Uitbreiding 2

Begin met de Rekening basisklasse abstract te maken.

De bedoeling van deze uitbreiding is dat elke rekening een code kan genereren. Deze volgen een ander stramien afhankelijk van het type van de rekening:

- ZichtRekening: ZR-maxKredietOnderNul-rekeningNr (ZR-100.00-123456)
- SpaarRekening: SR-rekeningNr-aangroeIntrest (SR-123456-2.00)

Voorzie in de juiste klasse de methode GeefCode(). Denk goed na op welke manier je deze klasse implementeert.

Print in de RekeningApplicatie voor elke Rekening die je terug krijgt uit de RekeningenRepository via de GeefRekeningen() functie de code uit.

Oefening 1 - Uitbreiding 3

Maak een nieuwe interface aan die IBeheersKost heet. Deze heeft één getter voor de property JaarlijkseKost.

Verder maken we nog een nieuwe klasse Kluis aan (implementeert interface IBeheersKost).

Deze heeft volgende readonly attributen: houder(String) en kluisnummer(int). Alsook een constructor waarin bij het aanmaken van een kluis-object al zijn attributen worden opgevuld.

Kluis heeft ook een eigen implementatie van de ToString methode, deze geeft het kluisnummer en houder als String terug

De jaarlijkse kost voor het houden van een kluis is vast, namelijk €29.

De bestaande klasse Rekening implementeert ook de interface IBeheersKost. Bekijk hoe je dit op een logische manier kan implementeren.

Het berekenen van de beheerskosten voor de verschillende rekeningtypes doen we als volgt:

- Zichtrekening: 1% van de positieve waarde van |maxKredietOnderNul|
- Spaarrekening: indien saldo < 300, dan 5% van (300 - saldo) anders 0

Maak een BeheersKostenRepository aan, gelijkaardig aan de RekeningenRepository. Deze bevat ook een methode die wat data terug geeft. Voorzie hier opnieuw een functie die een lijst terug geeft met volgende objecten:

1. Kluis: houder=Tania, kluisNummer=100
2. Zichtrekening: houder=Jan, rekeningNummer=123456700082, saldo=1500, maxKredietOnderNul=-2000
3. Kluis: houder=Steve, kluisNummer=250
4. Spaarrekening: houder=Sandra, rekeningNummer=123456780009, saldo=1250, intrest=5

Maak nu een implementatie die een output voorziet volgens onderstaand formaat (zonder kleuren). Een deel van de implementatie voor deze output ontbreekt nog, probeer dit zelf aan te vullen.

```
k Luisnr = 100, houder = Tania
Detail: k Luisnr = 100, houder = Tania
Kost: 29.0

zichtrekening, houder = Jan
Detail: ZichtRekening met rekeningnummer 123-4567000-82
staat op naam van Jan
en bevat 1500,00 euro. Max krediet onder nul = -2000,00
Kost: 20.0

k Luisnr = 250, houder = Steve
Detail: k Luisnr = 250, houder = Steve
Kost: 29.0

spaarrekening, houder = Sandra
Detail: SpaarRekening met rekeningnummer 123-4567800-09
staat op naam van Sandra
en bevat 1250,00 euro. Aangroeiintrest = 5,00%
Kost: 0.0
```