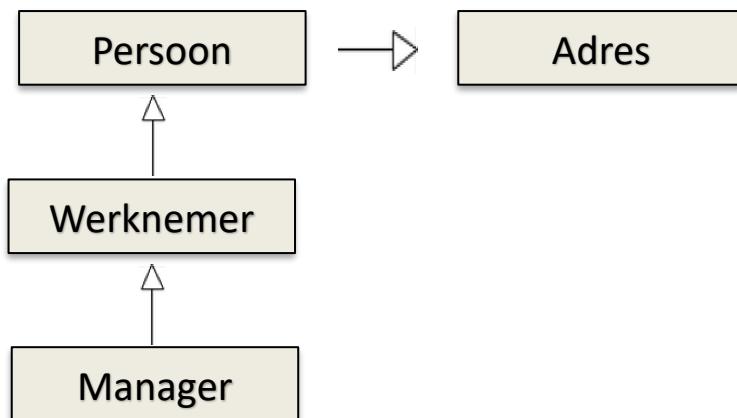


JAVA PROGRAMMING LANGUAGE

Oefening Overerving

Maak een Java Project aan met de naam Overerving.



klasse Adres

1. Maak een klasse Adres met de volgende kenmerken: straat (String), nummer (String), postcode (int) en gemeente (String). De kenmerken mogen van buiten de klasse niet toegankelijk zijn.
2. Voorzie een lege constructor en een constructor om een adres aan te maken met alle kenmerken.
3. Voorzie de nodige getters en setters.
4. Overschrijf de `toString()` methode om een adres als volgt weer te geven:

Veldkant 39

2550 Kontich

klasse Persoon

1. Maak een klasse Persoon met de volgende kenmerken: naam (String), voornaam (String) en adres (Adres). De kenmerken mogen van buiten de klasse niet toegankelijk zijn.
 2. Voorzie een constructor om een persoon aan te maken met volgende parameters: naam, voornaam, achternaam, straat, nummer, postcode en gemeente.
- Voorzie een tweede constructor om een persoon aan te maken door naam, voornaam, en

JAVA PROGRAMMING LANGUAGE

adres mee te geven.

Roep op een zinvolle manier 1 constructor op vanuit de andere.

3. Voorzie de nodige getters en setters.
4. Maak een methode voegVoornamenToe() om in 1 keer meerdere voornamen aan een persoon te kunnen toevoegen, vb p.voegVoornamenToe("An", "Els", "Kim");
Alle voornamen worden toegevoegd aan de String voornaam met telkens een spatie tussen.
5. Voorzie een methode toString() die de persoonsgegevens als volgt teruggeeft:

An Els Kim Van Brussel

Veldkant 39

2550 Kontich

klasse Werknemer

1. Maak een klasse Werknemer die afgeleid is van Persoon.
2. Een Werknemer heeft buiten een naam, voornaam en adres, ook een functie en een salaris.
3. Een Werknemer kan op 2 manier aangemaakt worden:
 - a. door enkel naam, voornaam en adres mee te geven: het salaris wordt ingesteld op 1800 en de functie op "algemeen bediende".
 - b. door naam, voornaam, adres, salaris en functie mee te geven.
4. Roep op een zinvolle manier vanuit de ene constructor de andere op.
5. Voorzie getters en setters voor alle eigenschappen.
6. Overschrijf de methode toString(). Zorg dat alle gegevens afgedrukt worden.
7. Het minimum-salaris is 1000. Maak hiervoor een klasse-variabele aan en zorg ervoor dat een salaris nooit lager kan zijn dan het toegestande minimum. Zorg ook dat je dit minimum salaris kan opvragen.
8. Doe het nodige om te weten hoeveel Werknemer-objecten gecreëerd werden.

klasse Manager

1. Maak een klasse Manager die afgeleid is van de klasse Werknemer.
2. Buiten een naam, voornaam, functie en salaris heeft een manager een bonus.
3. Voorzie 2 constructors:
 - a. door een waarde mee te geven voor naam, voornaam, adres, salaris en functie. De bonus wordt ingesteld op 50.
 - b. door een waarde mee te geven voor alle eigenschappen.
4. Voorzie een get- en set-methode om de bonus te kunnen opvragen en wijzigen.
5. Herdefinieer de methode getSalaris. Bij het salaris van de manager wordt de bonus bijgeteld.

JAVA PROGRAMMING LANGUAGE

6. Doe het nodige om te weten hoeveel Manager-objecten gecreëerd werden.
7. Voorzie een methode `getProcAandeelManagers()` die het procentueel aandeel van de managers in het totaal aantal werknemers teruggeeft (met 2 decimalen).

Maak een klasse `WerknemerApp` met een `main`-methode waarin je 2 `Werknemer`-objecten en minstens 1 `Manager`-objecten aanmaakt en alle methoden uittest.

Let zeker op het volgende:

- worden de juiste default-waarden ingesteld wanneer gebruik gemaakt wordt van de constructors waarbij niet alle waarden voor de eigenschappen meegegeven worden?
- is het loon bij niemand lager dan het minimumloon?
- Wordt bij managers het juiste loon afgedrukt?
- klopt het procentueel aandeel van de managers tov het totaal aantal werknemers?