

Oefeningen op factory patterns

A. Simple Factory

In een fictieve hogeschool zijn er studenten.

Zoals alle mensen hebben deze studenten regelmatig dorst, en drinken dan iets. Wat ze drinken hangt af van de omstandigheden.

- iets met cafeïne (cola);
- Of iets met alcohol (bier).

Pas het Simple Factory pattern toe op deze situatie:

- Maak een familie van dranktypes. Elke drank heeft een naam, en een methode getNaam().
- Maak een drankfactory. De factory past het simple factory pattern toe. De factory methode van de factory heeft een argument van type string dat bepaalt welk soort drank er gemaakt moet worden.
- Maak een klasse Student. Een student heeft een naam en een methode getNaam(). Een student heeft ook methodes DrinkCafeine() en DrinkAlcohol(). In die methodes wordt de factory gebruikt om een nieuwe drank te maken. De naam van die drank wordt dan afgedrukt.

Je krijgt alvast de main-methode:

```
public class OefeningSimpleFactory {  
    public static void main(String[] args) {  
        DrankFactory drankfactory = new DrankFactory();  
        Student student = new Student("Jan Janssens", drankfactory);  
  
        student.DrinkCafeine();  
        student.DrinkAlcohol();  
    }  
}
```

En dit is de verwachte output op de console:

```
Student Jan Janssens drinkt een cola  
Student Jan Janssens drinkt een bier
```

B. Factory Method

Het vorige voorbeeld wordt aangepast en uitgebreid. Er zijn nu verschillende soorten studenten:

- Clubleden, die houden van vertier;
- Boomknuffelaars, die houden van de natuur;
- Gamers, die houden van gamen.

Elk soort student heeft een eigen voorkeur voor drank voor overdag en voor 's avonds.

Clubleden drinken altijd bier, zowel overdag als 's avonds.

Boomknuffelaars drinken overdag wortelsap en 's avonds lindethee.

Gamers drinken overdag cola, en 's avonds RedBull.

Pas het Factory Method pattern toe op deze situatie:

- Breid de familie van dranktypes uit.
- Maak een familie van studententypes.
Een student heeft een naam en een methode getNaam(). Een student heeft ook methodes DrinkOverdag() en DrinkAvond().
- Bekijk het schema op blz 239. Hoe passen de dranken en de studenten in dat schema?
- Werk de toepassing van het pattern uit.

Dit is de main-methode:

```
public class OefeningFactoryMethod {  
    public static void main(String[] args) {  
        Student donkergroen = new BoomKnuffelaar("Dirk Donkergroen",  
groenFactory);  
        Student iovivat = new ClubLid("Johanna Vat", bierFactory);  
        Student wiebelaar = new Gamer("Henk Headshot",  
cafeineFactory);  
  
        donkergroen.DrinkOverdag();  
        iovivat.DrinkOverdag();  
        wiebelaar.DrinkOverdag();  
  
        donkergroen.DrinkAvond();  
        iovivat.DrinkAvond();  
        wiebelaar.DrinkAvond();  
    }  
}
```

En dit is de verwachte output op de console:

```
Student Dirk Donkergroen drinkt overdag een wortelsap  
Student Johanna Vat drinkt overdag een bier  
Student Henk Headshot drinkt overdag een Cola  
Student Dirk Donkergroen drinkt 's avonds een lindethee  
Student Johanna Vat drinkt 's avonds een bier  
Student Henk Headshot drinkt 's avonds een Redbull
```