

OOP1 – practicumopdracht 1

BSA Monitor 3.0, deel 1

Inleiding

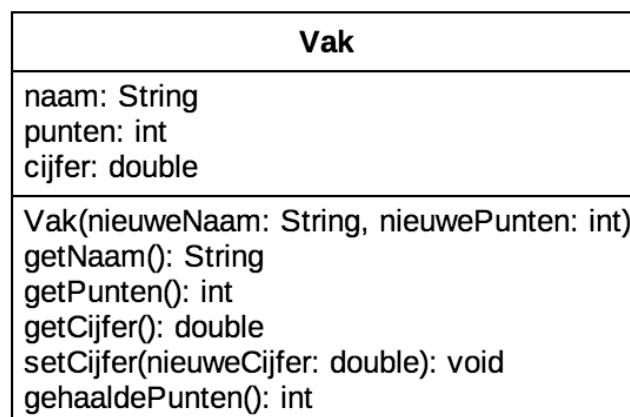
De meesten van jullie zullen de “BSA Monitor” opdracht van week 3 (1.0) en week 6 (2.0) nog wel herinneren. Dit is echt de allerlaatste keer dat we ermee aan de slag gaan.

Het vervelende aan deze opdracht in week 3 was dat je grote delen van je programma moest herhalen. Dat hebben we opgelost in week 6. Het vervelende aan de opdracht in week 6 was dat je allemaal losse arrays gebruikte van variabelen die eigenlijk bij elkaar hoorden (vakNamen, vakPunten, vakCijfers). Dat gaan we nu oplossen.

Ondertussen hebben we geleerd dat je ook een class kunt maken waarin je informatie bij elkaar zet die bij elkaar hoort (de attributen). Met onze kennis over classes en objecten kunnen we het programma verbeteren: we maken het korter, efficiënter en gemakkelijk uit te breiden met nieuwe vakken.

Deze applicatie bestaat uit twee opdrachten. Je hebt een week per opdracht om het te maken en in te leveren. Nadat je beide opdrachten af hebt, is BSA Monitor 3.0 af.

Het UML Class Diagram van de class Vak in stap 2 ziet er als volgt uit:



Stappenplan

1. Maak een nieuw Java project in IntelliJ genaamd “BsaMonitor3”.
2. Maak een nieuwe class genaamd Vak.
 - a. Deze class heeft de volgende attributen: naam, punten, cijfer. Tip: dit zijn de attributen die straks bij 1 object horen, dus het zijn geen arrays!
 - b. Maak een constructor Vak(String nieuweNaam, int nieuwePunten) die de nieuwe naam en de nieuwe punten van het vak kopieert in de attributen.
 - c. Maak een methode String getNaam() die de naam van het vak teruggeeft.

- d. Maak een methode `int getPunten()` die de punten van het vak teruggeeft.
 - e. Maak een methode `double getCijfer()` die het cijfer van het vak teruggeeft.
 - f. Maak een methode `setCijfer(double nieuwCijfer)` die het nieuwe cijfer van het vak kopieert in het attribuut.
 - g. Maak een methode `int gehaaldePunten()` die controleert of een vak is gehaald (dat wil zeggen: het cijfer is een voldoende) en die het behaalde aantal punten teruggeeft.¹
3. Test of dit alles werkt in de `main()` methode van class `BsaMonitor3`.
- a. Maak een object van type `Vak` aan door de constructor aan te roepen:
`Vak fys = new Vak("Fasten your Seatbelts", 12);`
 - b. Vraag de gebruiker om een cijfer voor dit vak en lees dit cijfer in.
 - c. Sla dit cijfer op bij het vak. Gebruik hiervoor de methode `setCijfer()` van het object `fys`.
 - d. Druk de volgende informatie van het vak af: naam, cijfer, gehaalde punten. Gebruik hiervoor de methoden `getNaam()`, `getCijfer()` en `gehaaldePunten()` van het object `fys`.
 - e. Kijk of het aantal behaalde punten klopt als je een voldoende of een onvoldoende invoert.

Input/Output

Zie hier twee voorbeelden van input/output van deze applicatie. Input die de gebruiker invoert is schuin en onderstreept.

```
Voer behaalde cijfer voor Fasten Your Seatbelts in: 5.5
Vak/Project: Fasten Your Seatbelts  Cijfer: 5.5  Punten: 12
```

```
Voer behaalde cijfer voor Fasten Your Seatbelts in: 5.4
Vak/Project: Fasten Your Seatbelts  Cijfer: 5.4  Punten: 0
```

Richtlijnen bij coderen (zie ook HBO-ICT code conventions)

- Zorg dat je naam en het doel van het programma bovenin staan (ICC #1).
- Gebruik de juiste inspringing (*indentation*) bij de lay-out (ICC #2) .
- Let op juist gebruik hoofdletters en kleine letters (ICC #3).
- Gebruik goede namen (ICC #4).
- Voeg waar nodig commentaar toe die inzicht geven in je code (ICC#7).
- Gebruik javadoc tag `@author`.

¹ Tip: je kan hiervoor een conditional operator gebruiken (zie §3.14 van Liang)