

Object Georiënteerd

OPDRACHT 1A

Maak een Auto klasse met de eigenschappen: merk en kleur.
En geeft de Auto als gedrag rijden() en remmen() met als status snelheid

Maak een Main klasse met een main methode
Instantieer een Auto als een rode Tesla als een mijnAuto.
Laat mijnAuto rijden en remmen.
Print de snelheid uit.

```
public class Auto {
    String merk;
    String kleur;

    int snelheid;

    public void rijden() {
        snelheid += 10;
    }
    public void remmen() {
        snelheid -= 10;
    }
}
```

```
public class Main {
    public static void
        main(String[] args) {
        Auto auto = new Auto();
        auto.merk = "Tesla";
        auto.kleur= "rood";

        auto.rijden();
        auto.remmen();
        System.out.println(
            auto.snelheid);
    }
}
```

OPDRACHT 1B

Maak een mijnAuto een grijze Suzuki. Laat hem drie keer rijden en print de snelheid

OPDRACHT 2

Maak de eigenschappen: merk en kleur private en genereer
getters en setters.

Rechtermuis-Source- Generate Getters and Setters
En wijzig de Main klasse zodat het weer werkt.

OPDRACHT 3

Geef de Auto een constructor met de eigenschappen: merk
en kleur

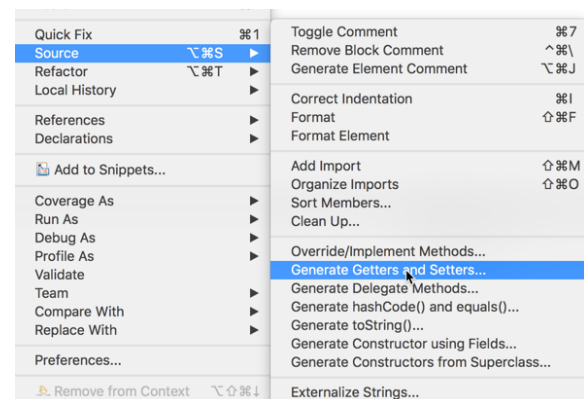
Rechtermuis-Source- Generate Constructor using Fields...
En wijzig de Main klasse zodat de constructor gebruikt wordt.

OPDRACHT 4

Maak de Auto klasse immutable
D.w.z. verwijder de setters.
En pas de Main klasse aan indien nodig.

OPDRACHT 5

Waarom krijgt de snelheid geen getters en setters?
Maak de snelheid ook private.
En genereer de toString methode
Rechtermuis-Source- Generate toString()...
En wijzig de Main klasse zodat het weer werkt.



OPDRACHT 6

maak een nieuwe klasse VruchtAuto, die is afgeleid van de Auto klasse.
Geef een VruchtAuto een extra eigenschap laadVermogen. Gebruik meteen getters en setters
Geef de VruchtAuto een constructor met de eigenschappen:merk en kleur
In de Main klasse maakt een blauwe Volvo onder de naam: truck1
En geef truck1 een laadVermogen van 1000.

OPDRACHT 7

Overschrijf de rijden en remmen methoden zodat dat ze langzamer rijden en remmen.
Laat eclipse de methode genereren met Rechtermuis-Source- Override/Implement Methods...
Zet de snelheid in Auto op protected.

```
public class VruchtAuto extends Auto {
    private int laadVermogen;

    public VruchtAuto(String merk,
                      String kleur) {
        super(merk, kleur);
    }
    public int getLaadVermogen() {
        return laadVermogen;
    }
    public void setLaadVermogen(
        int laadVermogen) {
        this.laadVermogen = laadVermogen;
    }
    @Override
    public void rijden() {
        snelheid += 5;
    }
    @Override
    public void remmen() {
        snelheid += 5;
    }
}
```

OPDRACHT 8

Maak een RaceAuto afgeleid van Auto.
En geef hem een eigenschap topSnelheid (met getters en setters)
Maak een rode Ferarri als mijnTweedeAuto

OPDRACHT 9

Maak een interface Voertuig met de methoden: rijden en remmen
Zorg dat de Auto klasse dit interface implementeert.

OPDRACHT 10

Maak een Fiets klasse die ook het Voertuig interface implementeert
Maak een grijze VanMook fiets als mijnFiets