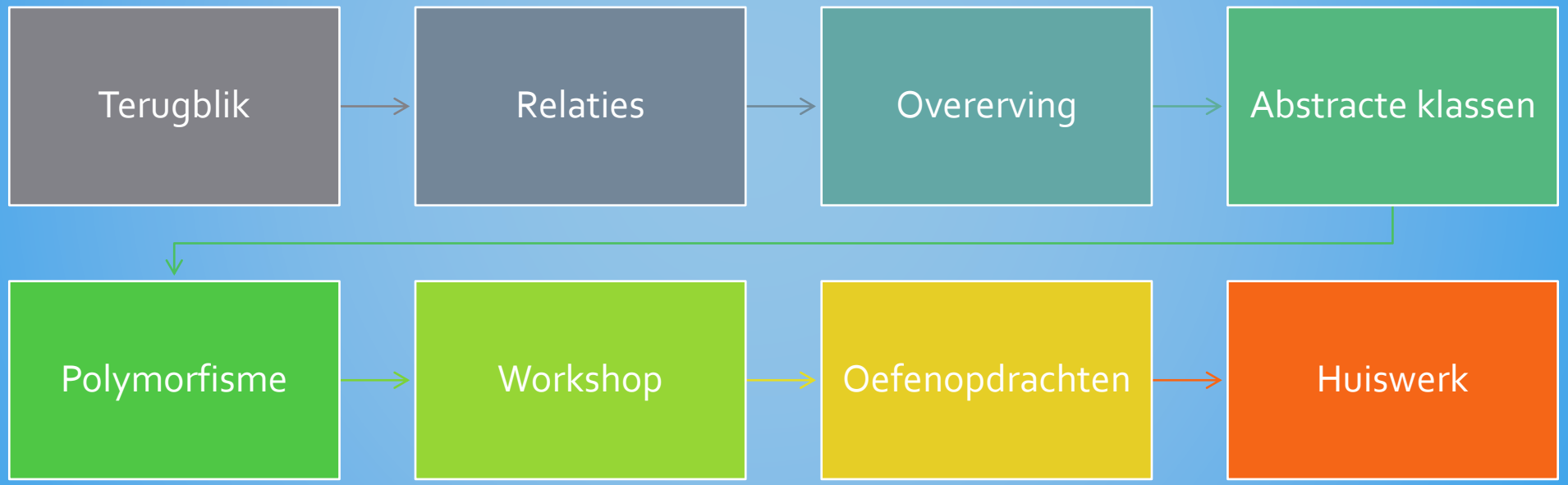
JAVA PROGRAMMEREN – LES 4:
RELATIES
OVERERVING
ABSTRACTIE
POLYMORFISME

Robert-Jan Elias

robert-jan.elias@novi-education.nl

NOVI

AGENDA


Terugblik
Relaties
Overerving
Abstracte klassen
Polymorfisme
Workshop
Oefenopdrachten
Huiswerk

NOVI

Arrays

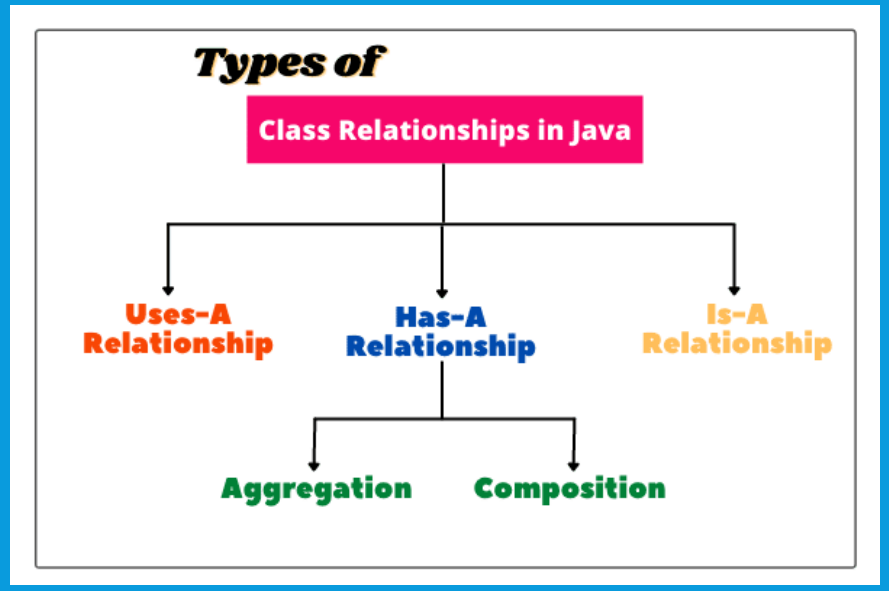
Collecties

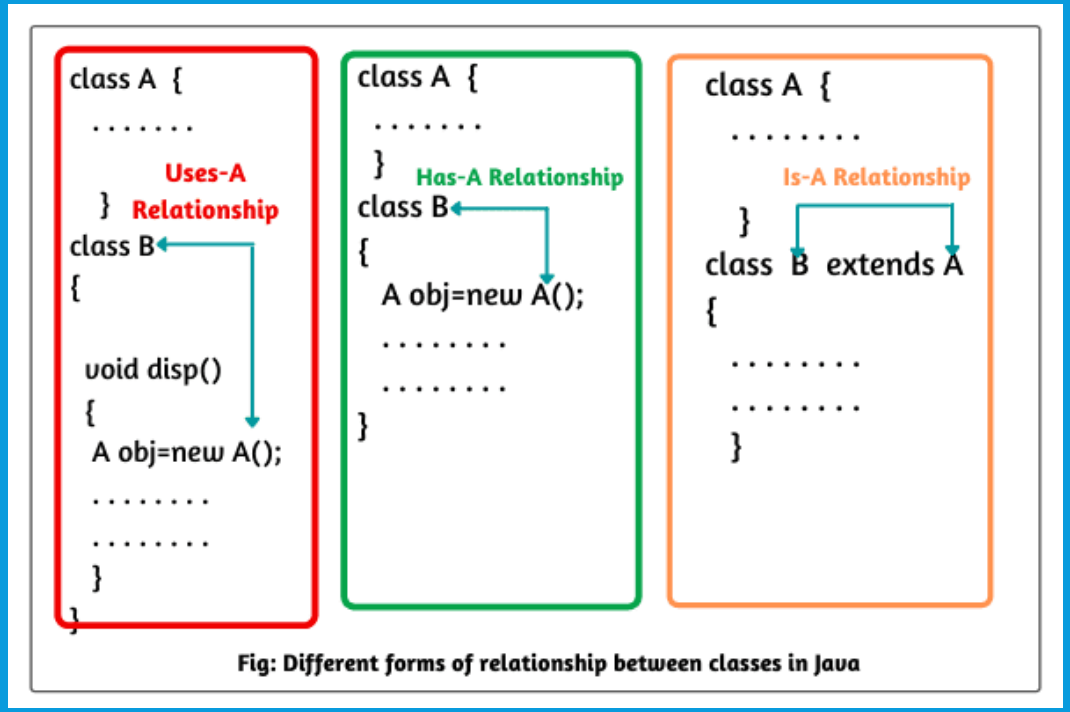
For- en while-lussen

TERUGBLIK

NOVI

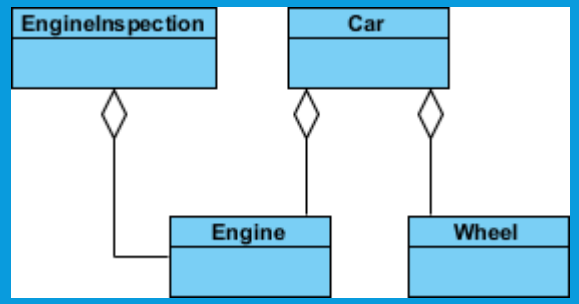
RELATIES TUSSEN KLASSEN



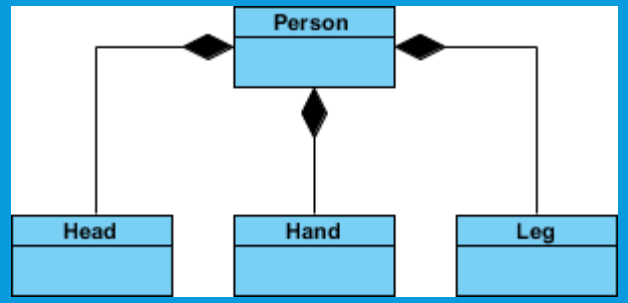


SOORTEN RELATIES (UML NOTATIE)

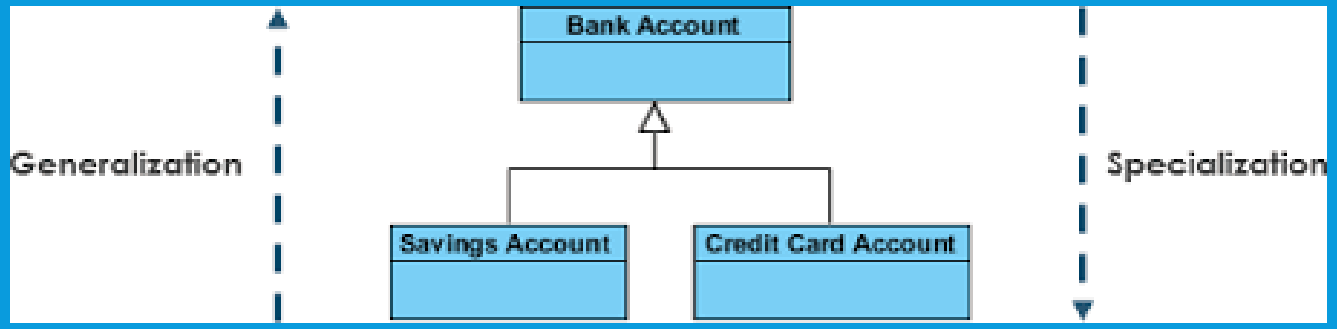
Aggregatie:



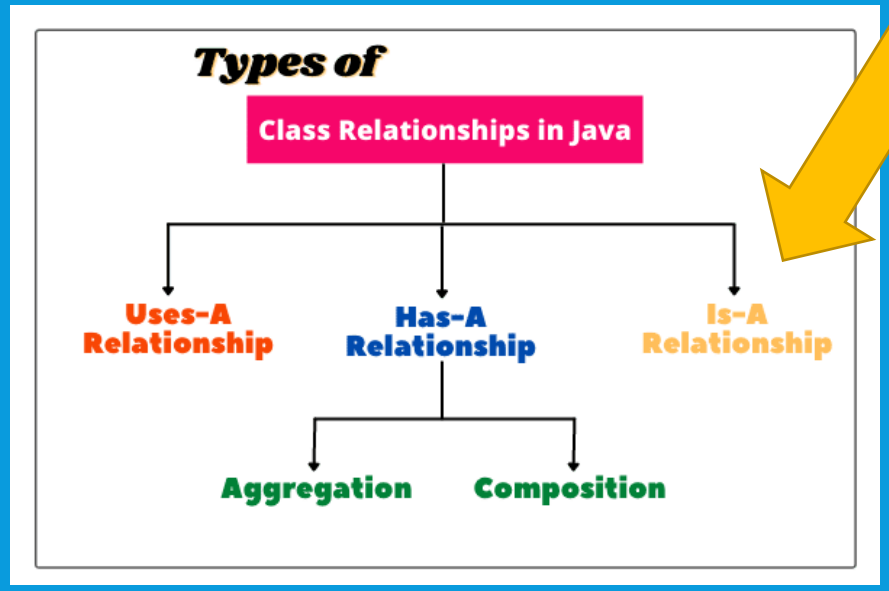
Compositie:

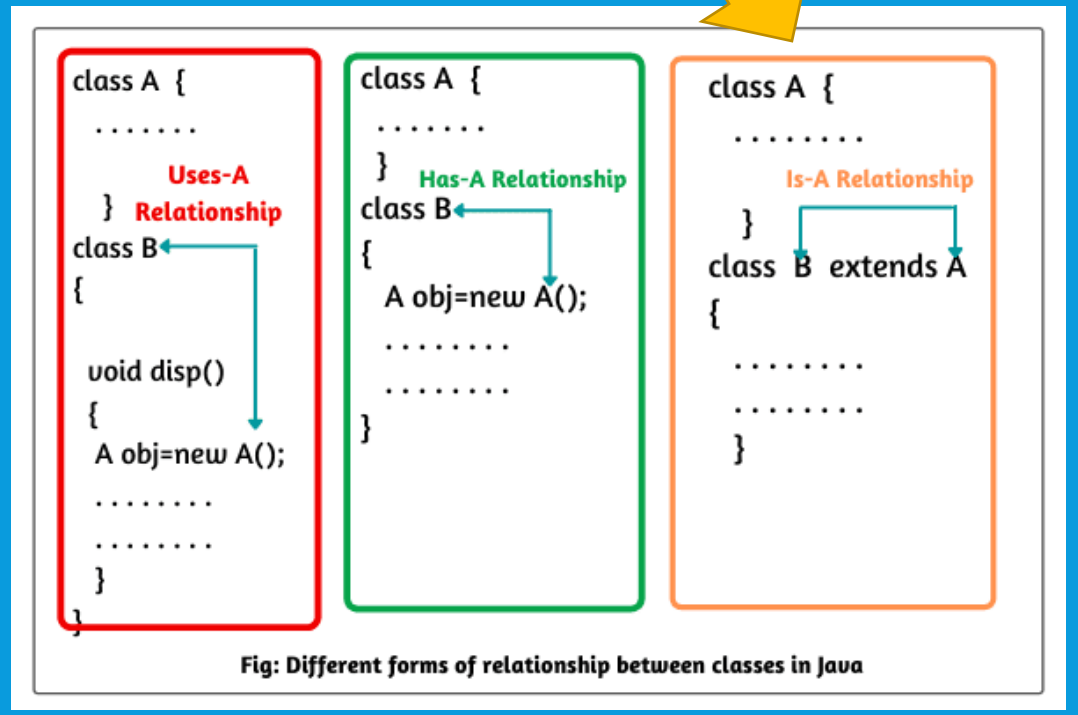


Overerving:



KLASSE OVERERVING



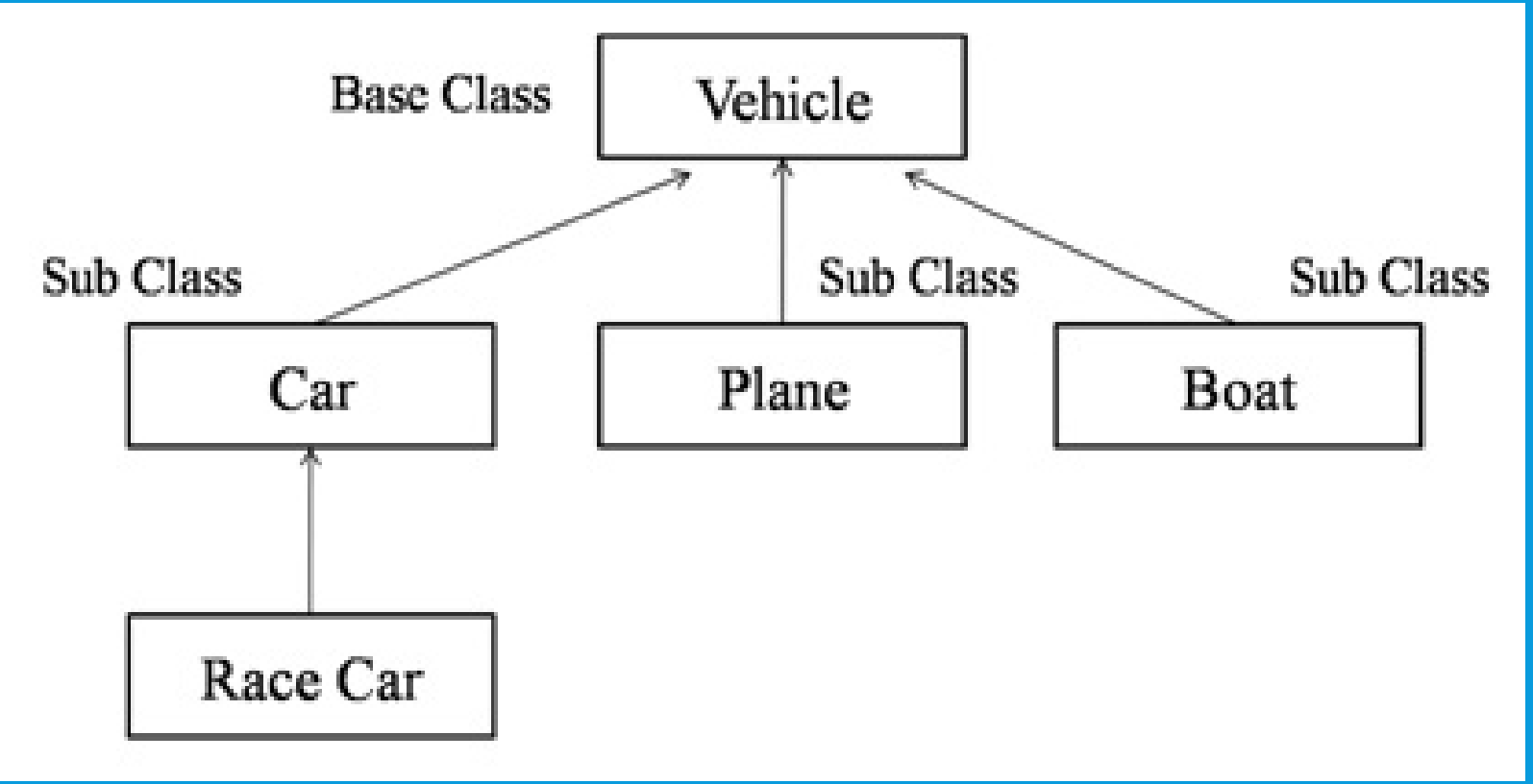


WAAROM KLASSE OVERERVING?

“We gebruiken bijvoorbeeld verschillende voertuigen om ons te verplaatsen. Ze hebben misschien bepaalde eigenschappen of functionaliteit gemeen. Maar ze hebben ook specifieke eigenschappen, zoals kunnen vliegen.

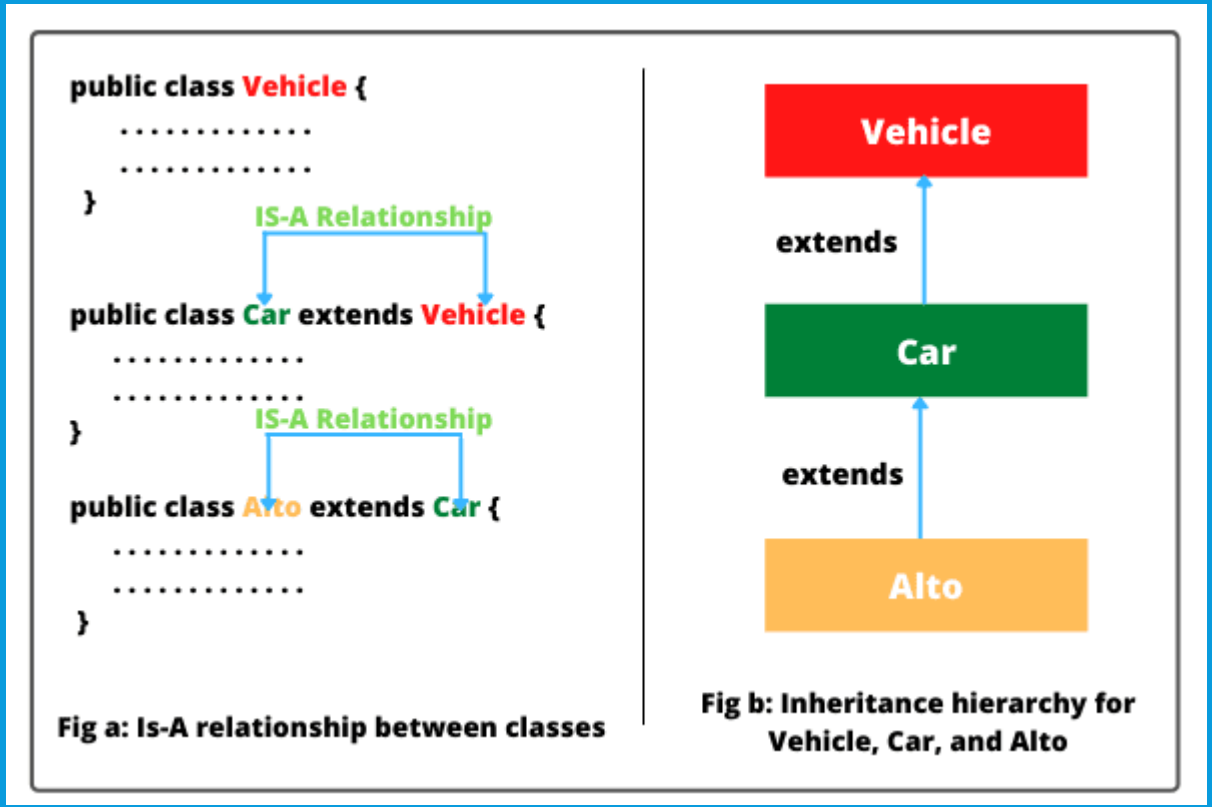
Je kunt de gemeenschappelijke eigenschappen en functionaliteiten per klasse apart definiëren, maar dat is weer véél dubbel werk en kan ook weer leiden tot meer fouten. Wat je kunt doen is voor deze klassen een overkoepelende klasse maken die alle eigenschappen en functionaliteiten kan bevatten en waar de onderliggende klassen gebruik van kan maken.”

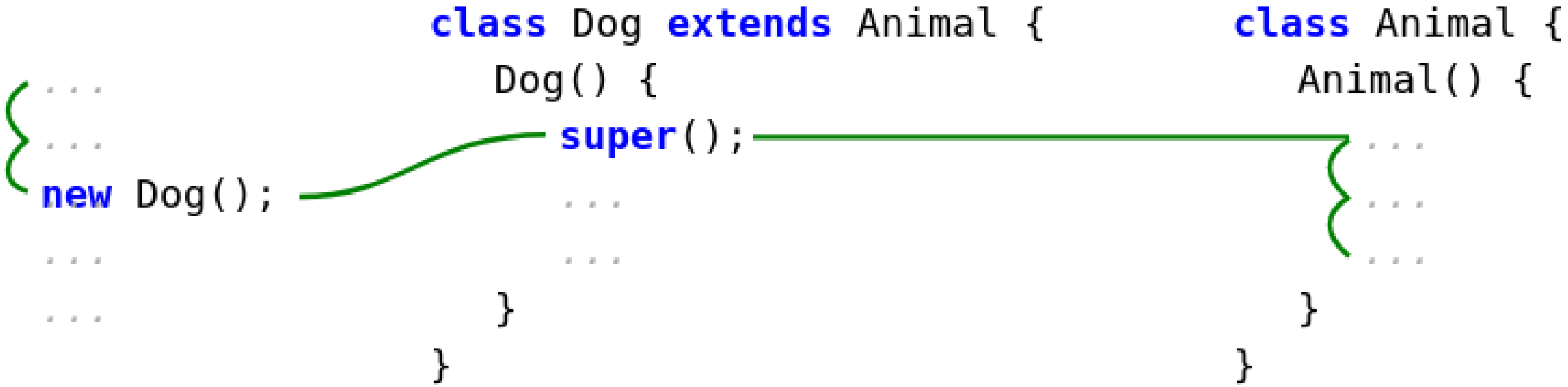
KLASSE OVERERVING



NOVI

JAVA INHERITANCE (= OVERERVING)





SUPER() INVOKES CONSTRUCTOR OF BASE CLASS

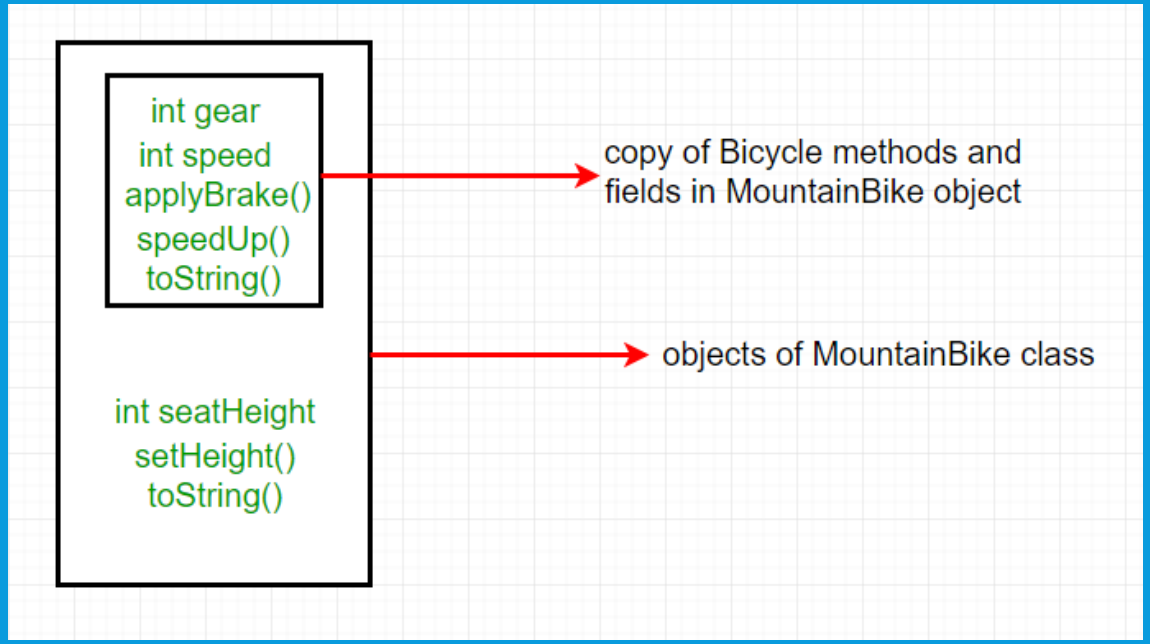
SUBKLASSE INSTANTIE

```java
// base class
class Bicycle {
    // the Bicycle class has two fields
    public int gear;
    public int speed;

    // the Bicycle class has one constructor
    public Bicycle(int gear, int speed)
    {
        this.gear = gear;
        this.speed = speed;
    }
```

```java
// derived class
class MountainBike extends Bicycle {

    // the MountainBike subclass adds one more field
    public int seatHeight;

    // the MountainBike subclass has one constructor
    public MountainBike(int gear, int speed,
                        int startHeight)
    {
        // invoking base-class(Bicycle) constructor
        super(gear, speed);
        seatHeight = startHeight;
    }
```

"new MountainBike(...)"

Object van subklasse:



ABSTRACTE KLASSEN

Abstracte klasse kan niet geïnstantieerd worden!

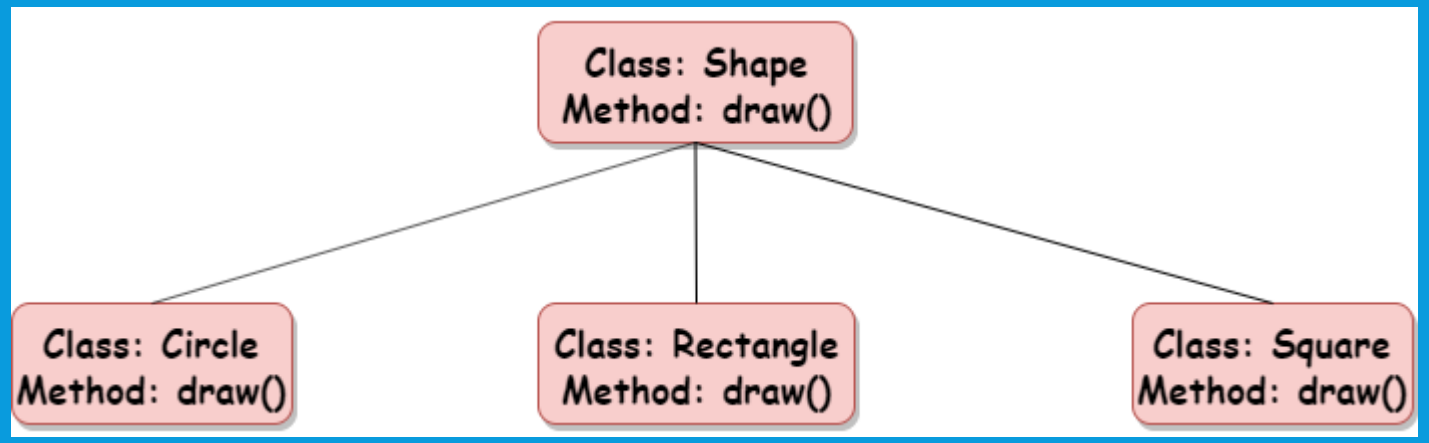
(dus Vehicle v = new Vehicle() mag niet)

```java
public abstract class Vehicle {
    int posX, posY;
    float direction;
    float speed;

    public abstract void drive();
}

class Car extends Vehicle {
    @Override
    public void drive() {
        System.out.println("Car driving");
        // ...
    }
}
```

NOVI

POLYMORFISME

- *"Perform the same action in many different ways"*



WORKSHOP “SHAPES”

- Wens: tekenen van allerlei vormen, zoals rechthoeken, cirkels, driehoeken etc.
- Gebruik klassen overerving

- Stappen:
- Technisch ontwerpje maken (in Visual Paradigm)
- Coderen (in IntelliJ)

NOVI

OEFENOPDRACHT "ANIMALS"

Download code van:

https://github.com/hogeschoolnovi/SD-BE-JP-Overerving-02

- Bestudeer opdracht eisen
- Maak base class Animal
- Voeg gemeenschappelijke eigenschappen en functies toe
- Bedenk klassen structuur (hoeveel lagen gebruik je?)
- Maak subclasses
- Instantieer objecten van de klassen en roep methods aan.

NOVI

HUISWERK

- Maak oefenopdracht “Animals” af

- Bestuderen Java Programmeren EdHub:
  - Hoofdstuk 5: Scope, access modifiers
  - Hoofdstuk 6: Static methods, overloading
  - Hoofdstuk 7: Interfaces

- Experimenteren op W3schools met:
  - Java Scope
  - Java Method Overloading