KU LEUVEN

Objectgericht Programmeren 1



joost.vennekens@kuleuven.be

Plaats in opleidingsaanbod

- Objectgericht ontwerpen
  - 35% Objectgericht ontwerpen
  - 65% OP1
- Informatiesystemen
  - 50% Databanken
  - 50% OP1

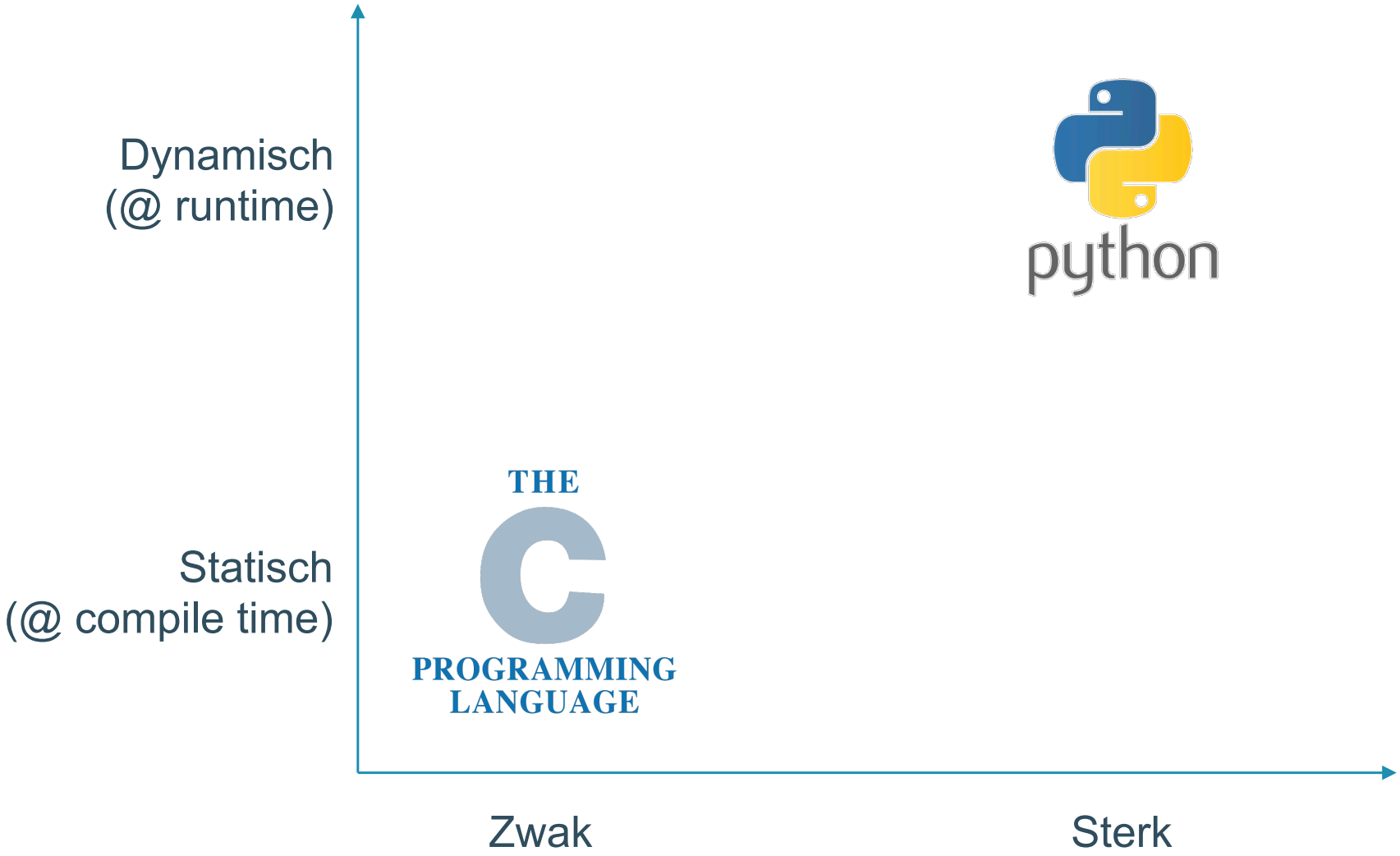
Evaluatie

- 40% Permanente evaluatie practica
- 60% Examen
  - 30% oefening
    - Gesloten boek
    - Op PC, zoals in practica
  - 30% theorie
    - Gesloten boek
    - Schriftelijk

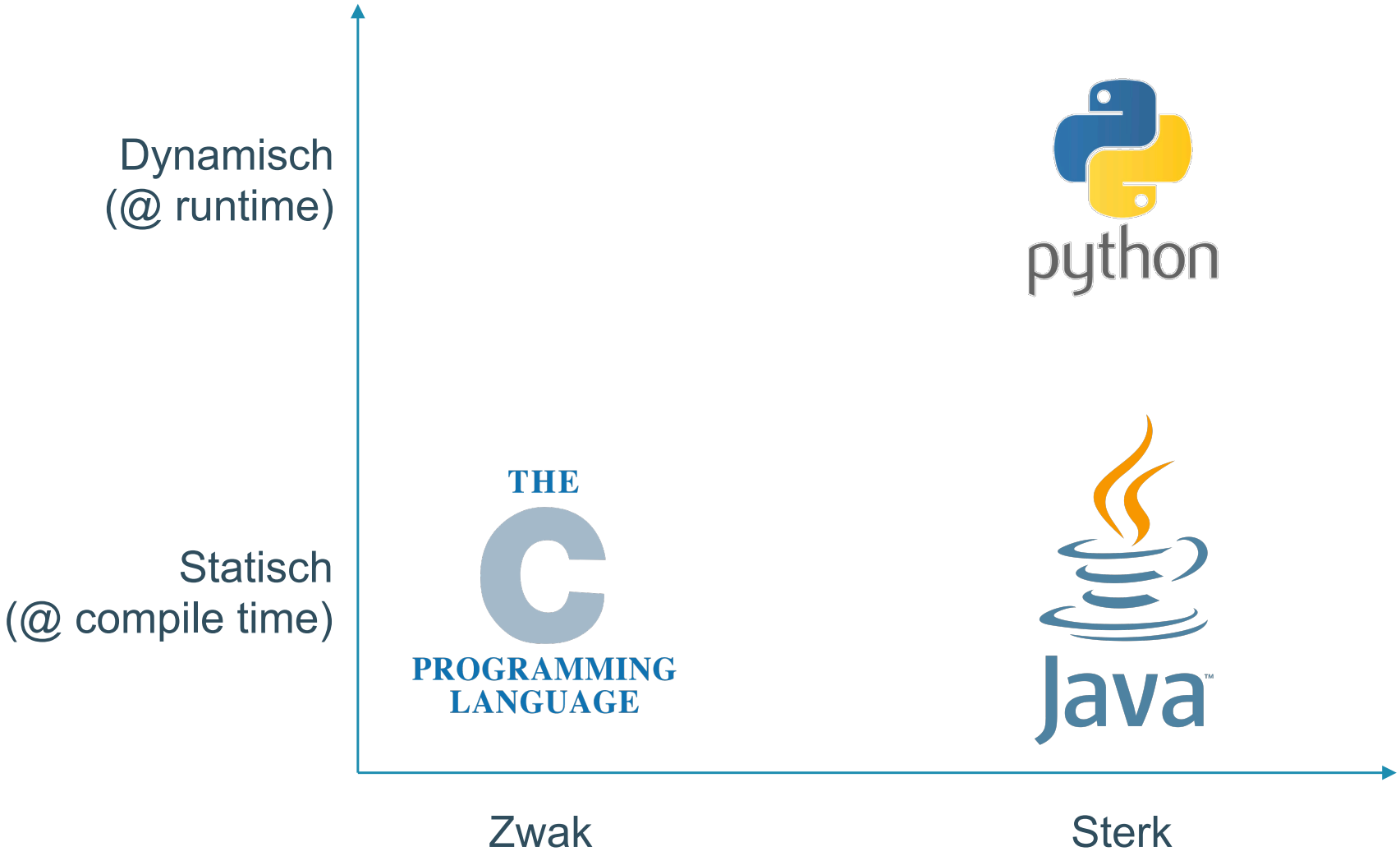
Inhoud: OOP

- Manier van denken over structuur van programma's
- In meer of mindere mate ondersteund door programmeertaal (C++, Python, …)
- In dit vak: Java

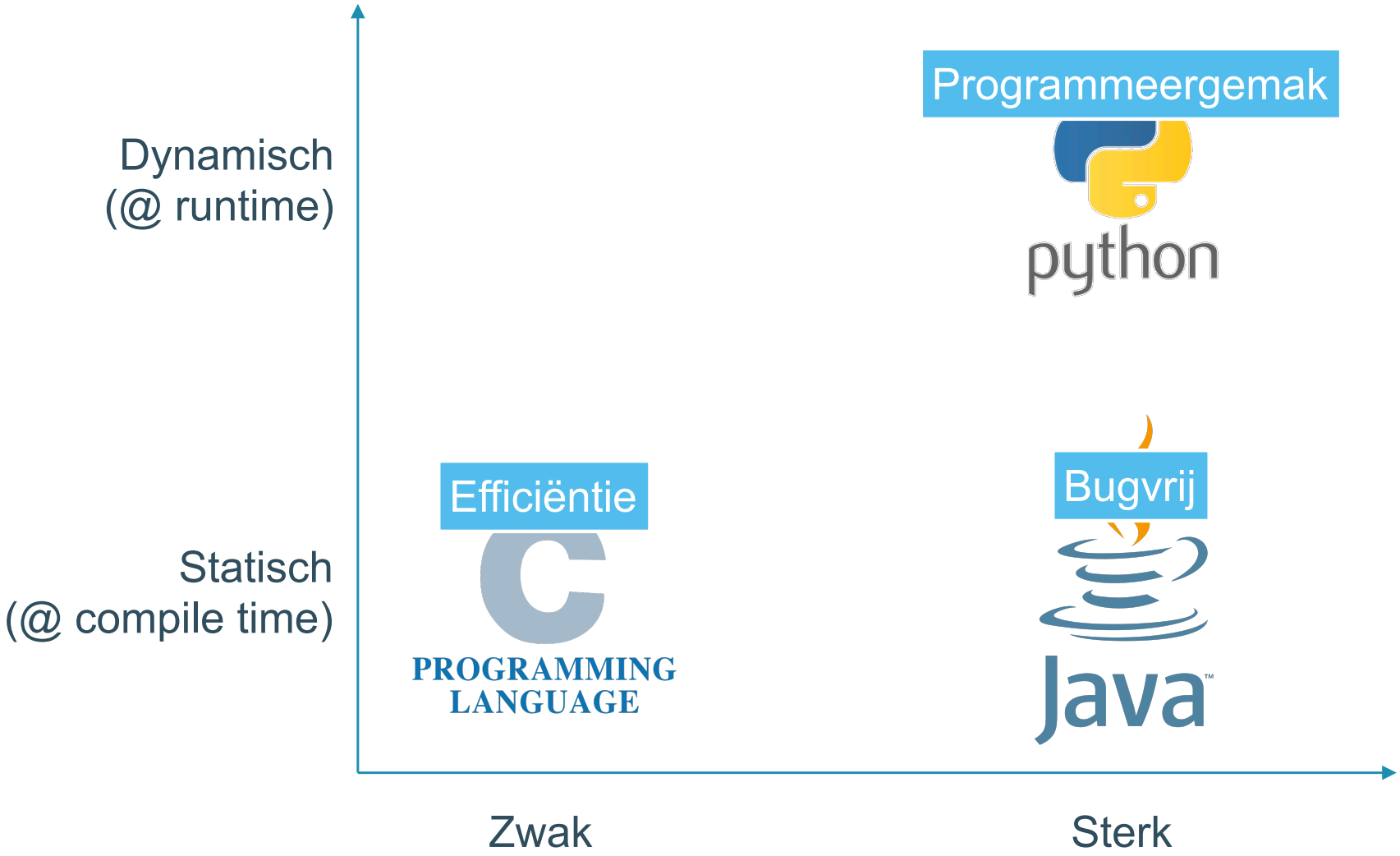
OO in Java


Dynamisch
(@ runtime)
Statisch
(@ compile time)
python
THE
C
PROGRAMMING
LANGUAGE
Zwak
Sterk

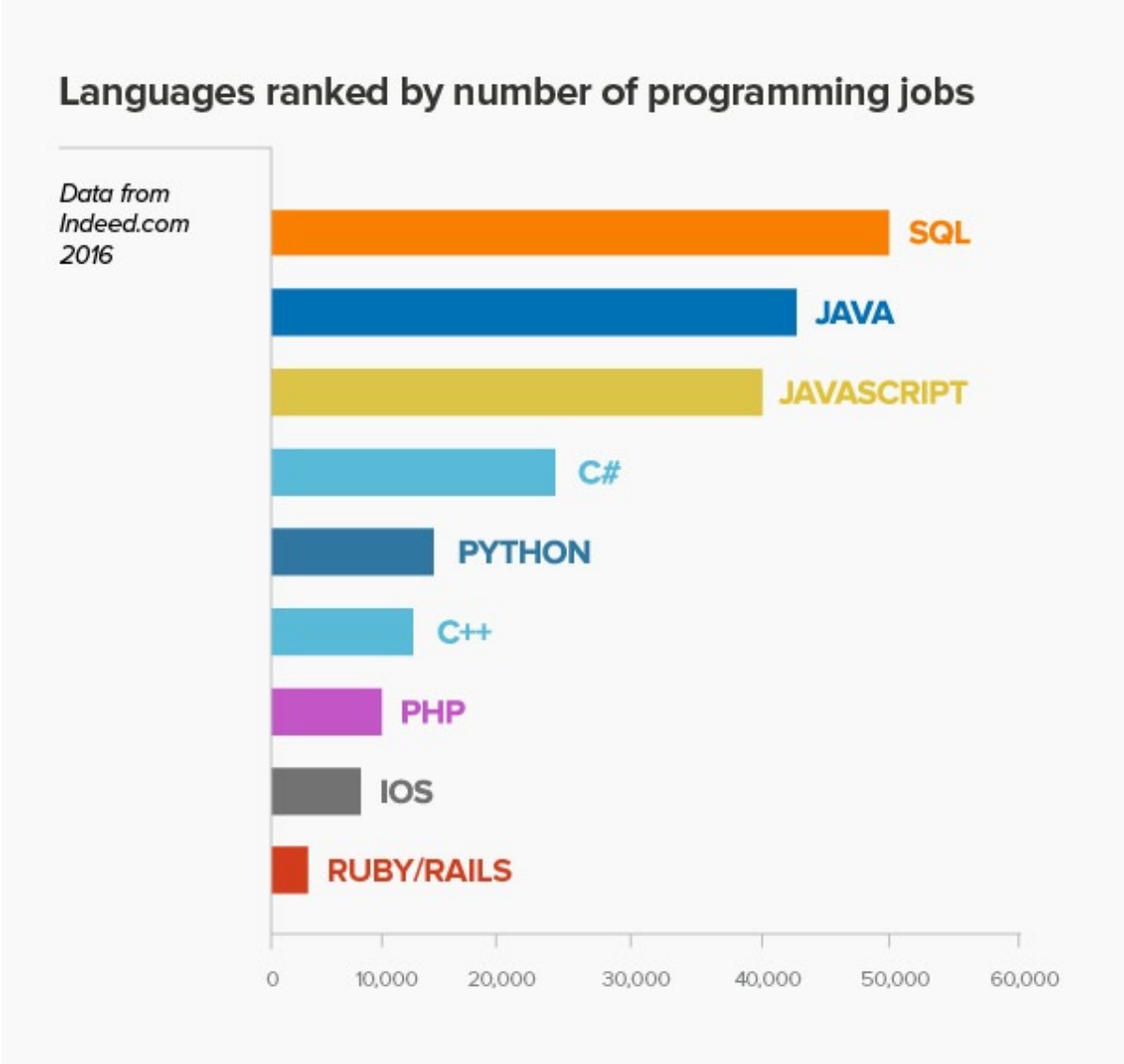
OO in Java


Dynamisch
(@ runtime)
Statisch
(@ compile time)
THE
C
PROGRAMMING
LANGUAGE
python
Java
Zwak
Sterk

OO in Java


Dynamisch
(@ runtime)
Statisch
(@ compile time)
Programmeergemak
python
Efficiëntie
C
PROGRAMMING
LANGUAGE
Bugvrij
Java
Zwak
Sterk

Populair in (grote) bedrijven



KU LEUVEN

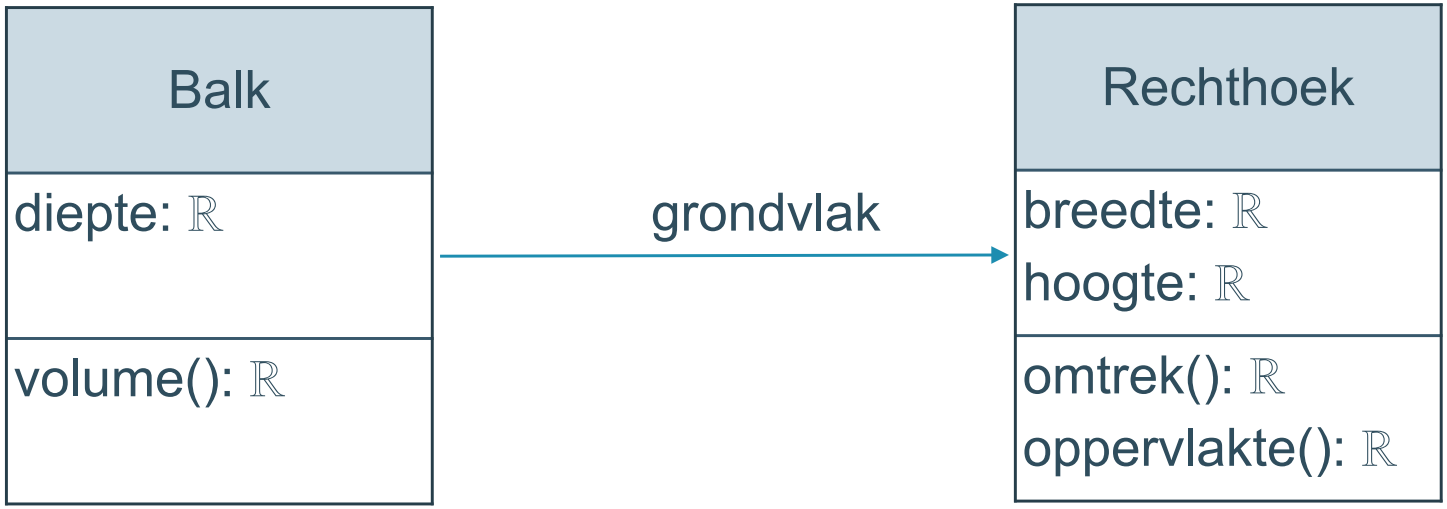
Voorbeeld

- Java programma bestaat uit:
  - Aantal klassen (en verder niets)
  - Elke klasse in gelijknamig bestand



Code + Stack/heap

Voorbeeld



Code + Stack/heap

Constructor

- Initialiseren van object
- Naam van de klasse
- Geen returntype (zelfs niet **void**)
- Indien afwezig: default constructor
- Andere constructor oproepen met **this**

Code aanpassen

Java is C-achtig

```
// Geheel
int geheelGetal;
long longerGetal;

// Komma
float drijvendeKomma;
double dubbelePrecisie;

// Booleaanse waardes
boolean waarOfFout;
```

```
// Tekst
char letter;
String stukjeTekst;

// Rijen
int[] rij;
int[][] matrix;
```

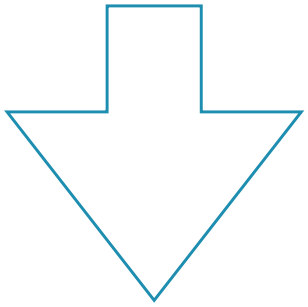
Lussen en tests

```
for (initializatie; test; stap) {
    ...
}

while (test) {
    ...
}

if (test) {
    ...
} else {
    ...
}
```

Object = Gegevens + Gedrag



Encapsulatie



Vb:twee soorten datums




```
class Datum {

    int sindsEenJanuari1980;

    Datum eenDagVroeger() {
        Datum gisteren = new Datum();
        gisteren.sindsEenJanuari1980 = sindsEenJanuari1980 - 1;
        return gisteren;
    }

}
```

```
Datum y = x.eenDagVroeger();
```

```
class DMJDatum {

    int dag;
    int maand;
    int jaar;

    Datum eenDagVroeger(){
        Datum gisteren = new Datum();
        if (dag > 1) { // Niet de 1e van de maand
            gisteren.dag = dag - 1;
            gisteren.maand = maand;
            gisteren.jaar = jaar;
        } else {
            if (maand > 1) { // 1e dag van maand
                gisteren.maand = maand - 1;
                gisteren.jaar = jaar;
                gisteren.dag = laatsteDag(gisteren.maand, gisteren.jaar);
            }
            else { // Nieuwjaar
                gisteren.maand = 12;
                gisteren.jaar = jaar - 1;
                gisteren.dag = 31;
            }
        }
        return gisteren;
    }
}
```




```
Datum y = x.eenDagVroeger();
```

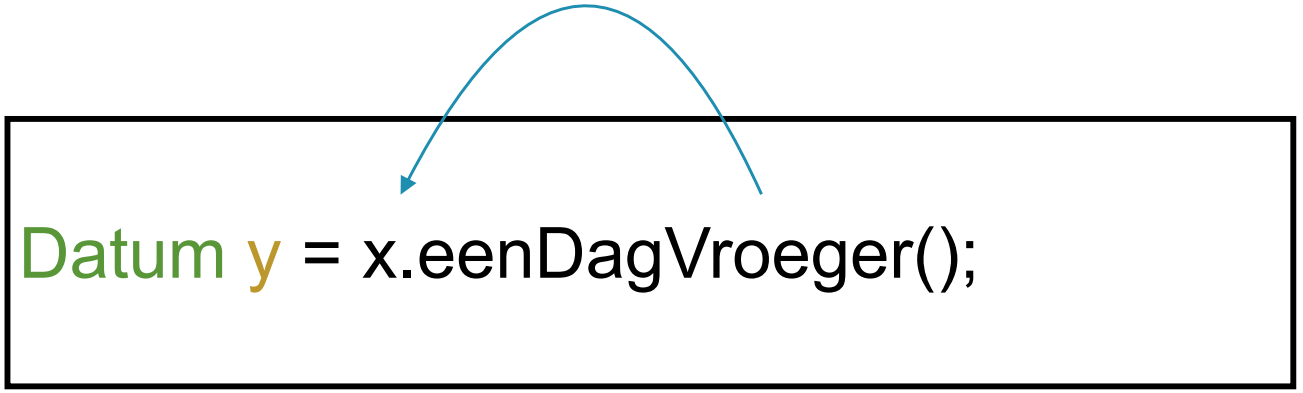
```
class Datum {

    ???

    Datum(int dag, int maand, int jaar) {
        ...
    }

    Datum eenDagVroeger() {
        ...
    }

}
```



Gegevens + gedrag

- Hoe gedrag geimplementeerd moet worden
- Hangt onafscheidelijk samen met hoe gegevens worden voorgesteld
- Dus gedrag en gegevens horen **samen**
- Object: gedrag + gegevens
  - Vergissingen vermijden
  - Flexibiliteit

Ondersteuning voor encapsulatie

- Toegansniveaus
  - private: voor niemand
  - niets: voor klassen uit zelfde pakket
  - protected: voor subklassen of uit hetzelfde pakket
  - public: voor iedereen



Rechthoek, Balk

```
class Datum {

    private int dag;
    private int maand;
    private int jaar;

    public Datum(int dag, int maand, int jaar) {
        ...
    }

    public Datum eenDagVroeger() {
        ...
    }

}
```

```
Datum x = new Datum(27,9,1980);
DMJDatum y = x.eenDagVroeger();
```

```
class Datum {

    private int sindsEenJanuari1980

    public void Datum(int dag, int maand, int jaar) {
        ...
    }

    public Datum eenDagVroeger() {
        ...
    }

}
```

```
Datum x = new Datum(27,9,1980);
DMJDatum y = x.eenDagVroeger();
```

Vb: complexe getallen

- Complex getal: reeel en imaginair deel

  c = a + bi
- Optellen: (a+bi) + (a'+b'i) = a+a' + (b+b')i
- Polaire voorstelling:

  magnitude *r en argument φ*
- $a = r \cos \varphi$ en $b = r \sin \varphi$

```
public class ComplexGetal {
    /** Complex getal: reeel + imaginair * i **/
    private double reeel;
    private double imaginair;

    public ComplexGetal(double r, double i) {
        reeel = r;
        imaginair = i;
    }
    public double getReeel() {
        return reeel;
    }
    public double getImaginair() {
        return imaginair;
    }
    public ComplexGetal plus(ComplexGetal ander) {
        return new ComplexGetal(ander.getReeel() + getReeel(),
                                ander.getImaginair() + getImaginair());
    }
}
```

```
public class ComplexGetal {

    private double magnitude;
    private double argument;

    public ComplexGetal(double r, double i) {
        magnitude = Math.sqrt(Math.pow(r,2) + Math.pow(i, 2));
        argument = Math.atan(i / r);
    }
    public double getReeel() {
        return magnitude * Math.cos(argument);
    }
    public double getImaginair() {
        return magnitude * Math.sin(argument);
    }
    public ComplexGetal plus(ComplexGetal ander) {
        return new ComplexGetal(ander.getReeel() + getReeel(),
                                ander.getImaginair() + getImaginair());
    }
}
```

Statische methodes

- Methodes die bij een **klasse** horen ipv. bij een **object**

```
public class Docent {
    public static Docent getRector() {
        return new Docent("Luc Sels");
    }
    private String naam;
    public Docent(String n) {
        naam = n;
    }
}
```

```
Docent d = new Docent();
d.getRector();

Docent.getRector();
```

Statische attributen

- Attributen die bij een **klasse** horen ipv. bij een **object**

```
public class Docent {
    public static Docent rector = new Docent("Luc Sels");
}
```

```
Docent d = new Docent();
d.rector;

Docent.rector;
```

Voorbeelden van static

```
public static void main(String[] args) { ... }

System.out.println(...);

public class System {
    public static PrintWriter out;
    ...
}

Math.sqrt(49);
```