

Methods

A programmer's journey.

A dark blue, diagonal shape that starts from the bottom left corner and extends towards the top right, covering the lower half of the slide. It has a smooth, slightly curved edge.

Inleiding

- ❏ Wat zijn methoden
- ❏ Waarom gebruiken we methodes
- ❏ Creëren van methodes
 - Demo
- ❏ Wat kan een methode hebben als return value
 - Demo
- ❏ Oefeningen

Wat is een methode

??

Een methode is een constructie om een aantal statements te groeperen zodat deze samen een functie vormen die kan aangeroepen worden en uitgevoerd wordt als een geheel

een methode:

- heeft dus een naam
- code die uitgevoerd moet worden
- een mogelijkheid om gegevens aan de methode mee te geven
- kan een resultaat teruggeven

Wij moeten nu de Java syntax leren om dit te doen

Methodes zullen we pas echt goed begrijpen op het moment dat we objecten bestuderen

voorbeeld: bepalen welk getal het grootste is

```
int result;  
if (num1>num2)  
    result = num1;  
else  
    result =num2;
```

Als we willen bepalen welk van twee getallen het grootste is, dan zouden we telkens *de statements* (links op deze slide) moeten schrijven

we zullen het stukje code (de statements) onderbrengen in een methode

```
int calculateHighest(int num1, int num2){  
    int result;  
    if (num1>num2)  
        result = num1;  
    else  
        result =num2;  
    return result;  
}
```

Daily routine

Dagelijkse instructies.

```
public void doDailyRoutine( ) {  
    opstaan;  
    bed opmaken;  
    ontbijten;  
    tanden poetsen;  
    met fiets naar trein rijden;  
    trein nemen naar Brussel;  
    werk;  
    pauze;  
    werk;  
    trein nemen naar Heist;  
    oma bezoeken  
    met fiets naar huis rijden;  
    avondmaal nuttigen;  
    ontspannen;  
    slapen;  
}
```

We gaan deze nu groeperen.

```
public void ochtendRoutine() {  
    opstaan;  
    bed opmaken;  
    ontbijten;  
    tanden poetsen;  
}
```

```
public void transport(String startPos, String eindPos) {  
    vertrekken op startPos;  
    verplaatsing;  
    aankomen op eindPos;  
}
```

```
public void avondRoutine() {  
    avondmaal nuttigen;  
    ontspannen;  
    slapen;  
}
```

Daily routine.

Dagelijkse instructies.

```
public String doDailyRoutine() {  
  
    jamie.ochtendRoutine( ) ;  
    jamie.transport("Ramskapelle","Brussel");  
    jamie.werkRoutine();  
    jamie.transport("Brussel","Ramskapelle");  
    jamie.avondRoutine( );  
  
    return "All done for the day";  
  
}
```

Nut van methodes

Reusable code

Als je (vaak) (bijna) dezelfde dingen wil doen kan je deze statements groeperen in een methode en deze methode oproepen wanneer je die nodig hebt.

Top-down programming

Je lost een groot probleem op door het in verschillende kleine problemen te verdelen.

Nut van methodes

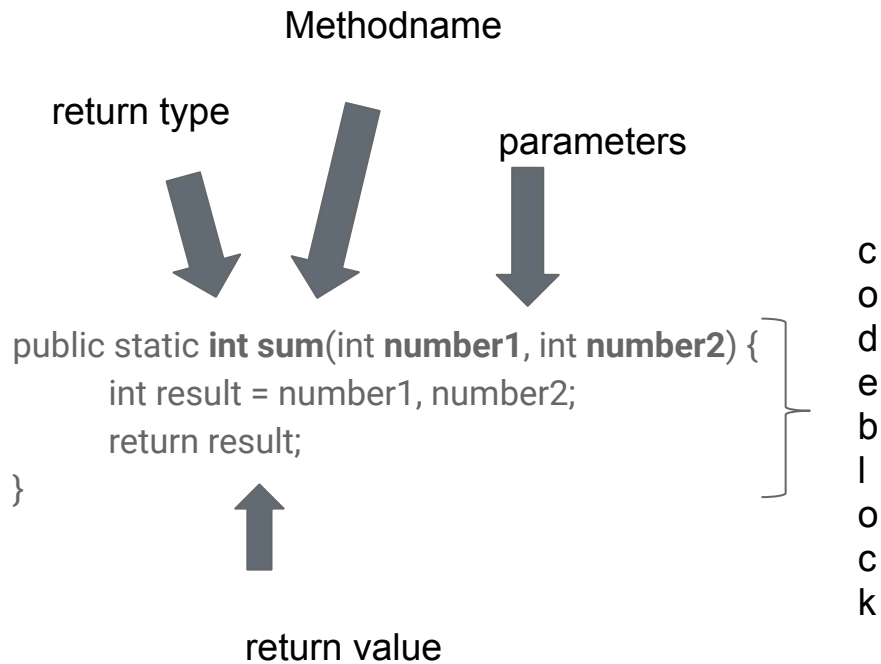
Creëren van conceptueel stukken

Schrijven van een methode die 1 actie vormt vanuit jouw mental view. Maakt het makkelijker om met je programma te werken.

Om te vereenvoudigen

Omdat lokale variabelen en statements niet van buiten de methode kunnen gezien worden, hierdoor zijn zij (en hun complexiteit) hidden voor andere delen van je programma. Dit voorkomt toevallige fouten en of verwarring.

Method signature



Gebruik van een
methode





