



Programming Basics

Hoofdstuk 4

Programmatie logica

DE HOGESCHOOL MET HET NETWERK

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook



Inhoud

1. Inleiding
2. Sequenties
3. Invoer en uitvoer
4. Keuzes (= selecties)
5. Herhalingen (=iteraties)
6. Samenvatting: programmeeralgoritmen



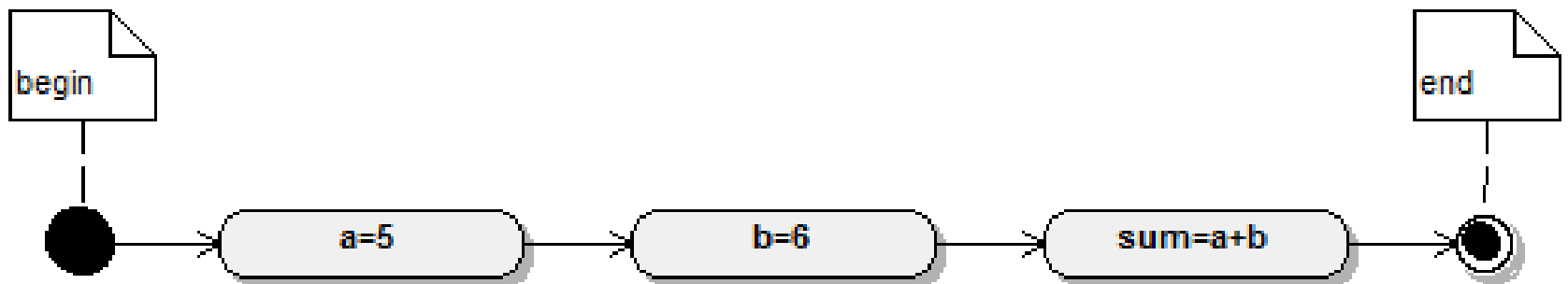
1. Inleiding

Een programma bestaat meestal uit:

- Sequenties: opeenvolgende instructies
- Keuzes: voorwaardelijke instructies
- Herhalingen: herhaaldelijke uitvoering van instructies
- Invoer: gegevens opvragen
- Uitvoer: gegevens ter beschikking stellen

2. Sequenties

Vb: Sequentie voor het optellen van 2 getallen



```
package be.px1.h4.opdracht;
```

```
public class Sum {
```

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

```
        int a = 5;
```

```
        int b = 6;
```

```
        int sum = a + b;
```

```
    }
```

```
}
```

= een sequentie

Puntkomma

= idem

De variabele a krijgt de waarde 5

Niet: a is gelijk aan 5!



```
int a;
```

```
a = 5;
```

```
int b = 6;
```

```
int sum = a + b;
```

a

5

b

6

→ sum

11

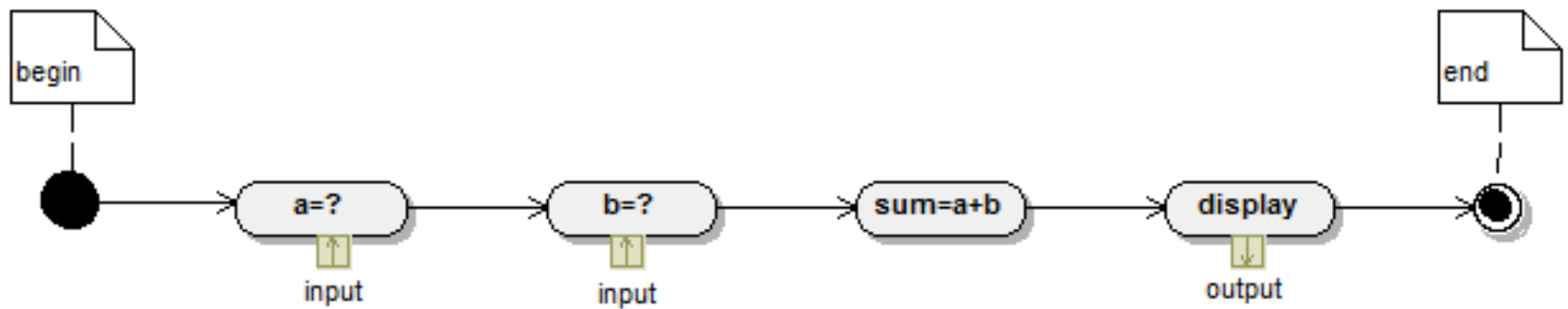
11



Opdracht 1: *Sequentie*

- Typ het voorbeeld in en voer uit
- Probeer het resultaat van de som (sum) eens te tonen (= output)

3. Invoer en uitvoer



```
package be.px1.h4.opdracht;  
import java.util.*;
```

```
public class Sum {  
    public static void main(String[] args) {  
        Scanner keyboard = new Scanner(System.in);
```

```
        System.out.println("Enter a number:";  
        int a = keyboard.nextInt(); // Input
```

= zelf gekozen naam

```
        System.out.println("Enter another number:");  
        int b = keyboard.nextInt(); // Input
```

```
        int sum = a + b;
```

= drukt boodschap af + NewLine

```
        System.out.println("The sum is " + sum); // Output
```

```
        keyboard.close();
```

// = commentaar

```
    }
```

```
}
```

Details van de code wordt in
volgende hoofdstukken duidelijk

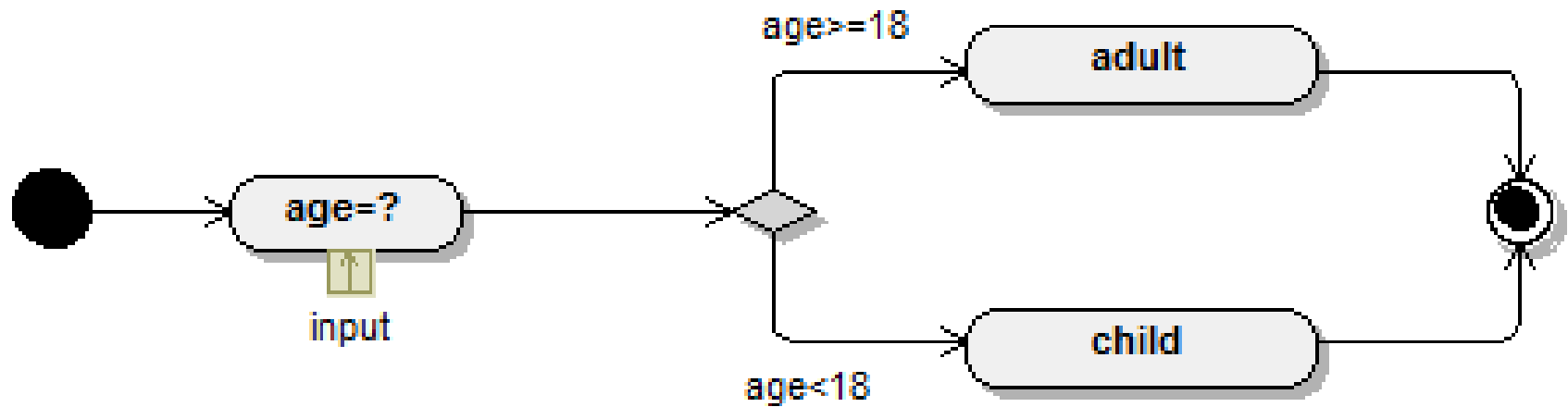


Opdracht 2: *Invoer en uitvoer*

Vervolledig het programma uit vorig voorbeeld met invoer en uitvoer. Bekijk het resultaat.



4. Keuzes: if ... else ...



```
package be.px1.h4.opdracht;
```

```
import java.util.*;
```

```
public class Age {  
    public static void main(String[] args) {  
        Scanner keyboard = new Scanner(System.in);
```

```
        System.out.println("Enter your age:");  
        int age = keyboard.nextInt();
```

```
        if (age >= 18) {  
            System.out.println("You are an adult");  
        } else {  
            System.out.println("You are a child");  
        }
```

```
        keyboard.close();
```

```
    }
```

```
}
```

= een selectie

if (condition) {...} else {...}

else {...} is optioneel



Geneste if ... else ...

```
if (age >= 18) {  
    System.out.println("You are an adult");  
} else {  
    if (age >= 10) {  
        System.out.println("You are a teenager");  
    } else {  
        if (age >= 2) {  
            System.out.println("You are a child");  
        } else {  
            System.out.println("You are a baby");  
        }  
    }  
}
```



De plaats van de accolades is belangrijk! Zorg verder voor een nette inspruing van de accolades (= een code convention). Dit verhoogt de leesbaarheid van het programma.



Source > Format: tracht te achterhalen
waarvoor dit commando dient!



Opdracht 3: *Selectie (if ... else ...)*

Maak bovenstaand programma in Java en voer uit

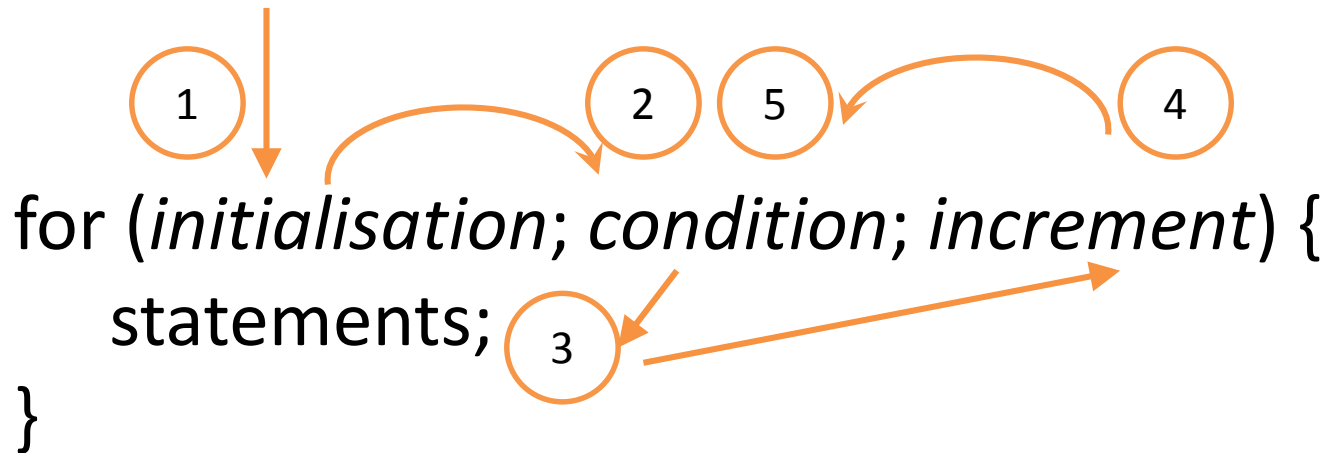


5. Herhalingen: for / while / do ... while

```
for (int count = 1; count < 5; count++) {  
    System.out.println(count);  
}
```

Output = ?

➔
1
2
3
4



Opdracht 4: *for*

1. Toon alle veelvouden van 10 die kleiner of gelijk zijn aan 50.

2.

```
for (int count = 1; count < 5; count++) {  
    System.out.println(count);  
    count = 5;  
}
```

Hoe vaak wordt de lus uitgevoerd? **1x**

➔ Verander een lusvariabele (*count*) van een for nooit in de body. In dat geval gebruik je beter een while (zie verder)




```
for (int count = 1; count < 5; count++) {  
    System.out.println(count);  
}
```

 = een iteratie

is equivalent met

```
int count = 1;  
while (count < 5) {  
    System.out.println(count);  
    count++;  
}
```

en

```
int count = 1;  
do {  
    System.out.println(count);  
    count++;  
} while (count < 5);
```

Output =

1

2

3

4



- Als het aantal stappen op voorhand gekend is gebruik je best een *for*!
- *while*: er wordt eerst getest. Afhankelijk van de test wordt de body wel/niet uitgevoerd
- *do ... while*: de body wordt minimaal 1x uitgevoerd. De test gebeurt achteraf.
- Vermits in ons vorig voorbeeld het aantal stappen (4 stappen; getallen van 1 t.e.m. 4) vooraf gekend is, gebruik je in dit voorbeeld best een *for*! Hier volgt een beter voorbeeld van de *while*.



Lees een rij getallen in. Zolang het ingelezen getal kleiner is dan 10 druk je het getal samen met het dubbel van dit getal af.

```
int getal;  
int dubbel;  
Scanner keyboard = new Scanner(System.in);  
  
getal = keyboard.nextInt();  
while (getal < 10) {  
    dubbel = getal * 2;  
    System.out.println("Het dubbel van " + getal + " is " + dubbel);  
    getal = keyboard.nextInt();  
}  
keyboard.close();
```

Opdracht 5: *while*

Lees een rij getallen in. Zolang het ingelezen getal groter is dan 100 druk je het getal samen met het 3-dubbele van dit getal af.



Tip: werk zoveel mogelijk met for, vervolgens zoveel mogelijk met while en vermijd de variant do ... while!!!

6. Samenvatting: programmeeralgoritmen

Een programma is

- een opeenvolging van instructies (sequenties)
- met eventueel keuzes (selecties)
- en herhalingen (iteraties)

om zo tot de oplossing van een bepaald probleem te komen

