

Controle Structuren in Java



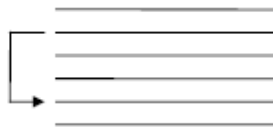
Klasgroep	1EO-ICT
Opleiding	Bachelor Elektronica-ICT
Lokaal	Groot Auditorium
Tijdstip	maandag lestijd 3
Docent	Katja Verbeeck
Contact	katja.verbeeck@odisee.be
Handboek	hfst 3 - p 81 - p 100

Control flow in java

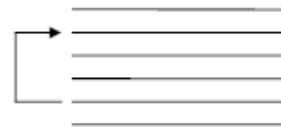
Binnen een methode worden instructies normaal sequentieel uitgevoerd:



Nieuw: een keuze maken
uit alternatieve stukken code



Nieuw: een stuk code
een aantal keer uitvoeren



De selecties:

- **if**-structuur
- **if/else**-structuur
- **meervoudige if/else**-structuur
- **switch**-structuur
- nesten van selectiestructuren

De iteraties:


- **while**-structuur
- **do/while**-structuur
- **for**-lus
- nesten van iteratiestructuren

Combinaties van controlestructuren


Een gekend aantal keer iets herhalen ... de for lus

Algemene vorm van een for-lus:

```
for (idx = waarde ; booleaanse uitdrukking ; idx aangepast) {  
    ...  
}
```



een teller bijhouden;
(declaratie) +
initialisatie



een stopconditie,
een vraag die je
beantwoordt met
true of false



de teller van waarde
veranderen

Intermezzo

Debuggen in Netbeans adhv een voorbeeld

Opgave:

[MinMaxKlinkers] Schrijf een programma waarbij de gebruiker N woorden intypt en het woord met het minst aantal klinkers en het woord met het meest aantal klinkers opnieuw uitprint naar het scherm. Bij ex aequo print je het eerste woord af. Het getal N definieer je als constante.

[MinMaxKlinkers2] Schrijf een programma waarbij de gebruiker N woorden intypt en het woord met het minst aantal VERSCHILLENDE klinkers en het woord met het meest aantal VERSCHILLENDE klinkers opnieuw uitprint naar het scherm. Bij ex aequo print je het eerste woord af. Het getal N definieer je als constante.

MinMaxKlinkers

```
// lees N woorden in en doe voor elk woord een
// klinkertelling
for(int i = 0; i < N; i++){
    System.out.println("Geef je volgende woord : ");
    String woord = scan.next();

    // de klinkertelling
    int aantal = 0;
    for(int j =0; j < woord.length(); j++){
        char kar = woord.charAt(j);
        if(klinkers.indexOf(kar) != -1)
            aantal++;
    }
    // is er een nieuw maximum of minimum bereikt ?
    ...
}
```

MinMaxKlinkers2

```
// lees N woorden in en doe voor elk woord een klinkertelling
for(int i = 0; i < N; i++){
    System.out.println("Geef je volgende woord : ");
    String woord = scan.next();
    String restKlinkers = klinkers;
    // de klinkertelling : opgelet je mag nu geen dubbels tellen !!
    int aantal = 0;
    for(int j =0; j < woord.length(); j++){
        char kar = woord.charAt(j);
        int index = restKlinkers.indexOf(kar);
        if(index != -1){
            aantal++;
            // verwijder de net gevonden klinker uit restKlinkers
            if (index == 0)
                restKlinkers = restKlinkers.substring(1);
            else if (index == woord.length() - 1)
                restKlinkers = restKlinkers.substring(0,index - 1);
            else restKlinkers = restKlinkers.substring(0,index) +
                               restKlinkers.substring(index + 1);
        }
    }
    // is er een nieuw maximum of minimum bereikt ?
    ...
}
```

Netbeans

- **Organiseert code in packages**
- **Apart output venster**
- **Bevat on-the-fly documentatie**
- **Laat toe om debug points te plaatsen en stap voor stap door de code te gaan**

On-the-fly documentatie

```
t min = Integer.MAX_VALUE;
```

```
ring minW java.lang.String
```

```
ring maxW
```

```
public char charAt(int index)
```

Returns the char value at the specified index. An index ranges from 0 to length() - 1. The first char value of the sequence is at index 0, the next at index 1, and so on, as for array indexing.

If the char value specified by the index is a [surrogate](#), the surrogate value is

```
int aan charAt(int index)
```

```
for(int chars()
```

```
    codePointAt(int index)
```

```
    if(codePointBefore(int index)
```

```
        codePointCount(int beginIndex, int endIndex)
```

```
    }  
    codePoints()
```

```
    compareTo(String anotherString)
```

```
    compareToIgnoreCase(String str)
```

```
if(aant concat(String str)
```

```
    contains(CharSequence s)
```

```
    contentEquals(CharSequence cs)
```

```
    contentEquals(StringBuffer sb)
```

```
    endsWith(String suffix)
```

```
if(aant equals(Object anObject)
```

```
    equalsIgnoreCase(String anotherString)
```

```
    getBytes()
```

```
    getBytes(Charset charset)
```

```
klinkers.
```

Instance Members: Press 'Ctrl+SPACE' Again for All Items

Debuggen in Netbeans

The screenshot shows the NetBeans IDE interface with the following components:

- Source Editor:** Displays the code for `MinMaxKlinkers2.java`. The code is as follows:

```
35
36 // lees N woorden in en doe voor elk woord een klinkertelling
37 for(int i = 0; i < N; i++){
38
39     System.out.println("Geef je volgende woord : ");
40     String woord = scan.next();
41
42     // de klinkertelling
43     int aantal = 0;
44     for(int j =0; j < woord.length(); j++){
45         char kar = woord.charAt(j);
46         if(klinkers.indexOf(kar) != -1)
47             aantal++;
48     }
49 }
```
- Variables Window:** Shows the current state of variables. The table below represents the data shown in the window:

Name	Type	Value
<Enter new watch>		
Static		
args	String[]	#262(length=0)
scan	Scanner	#263
max	int	0
min	int	2147483647
minWoord	String	""
maxWoord	String	""
i	int	0
woord	String	"eenvoudig"

- Output Window:** Shows the program's output and debug messages. The content is as follows:

```
debug:
Geef 5 woorden in :
Geef je volgende woord :
eenvoudig|
```

herhalen zolang de conditie waar is ... do/while of while lus

Algemene vorm van een `while`-lus:

```
while (booleaanse uitdrukking) {  
    ...  
}
```

Algemene vorm van een `do/while`-lus:

```
do {  
    ...  
} while (booleaanse uitdrukking);
```

Iteraties (while)

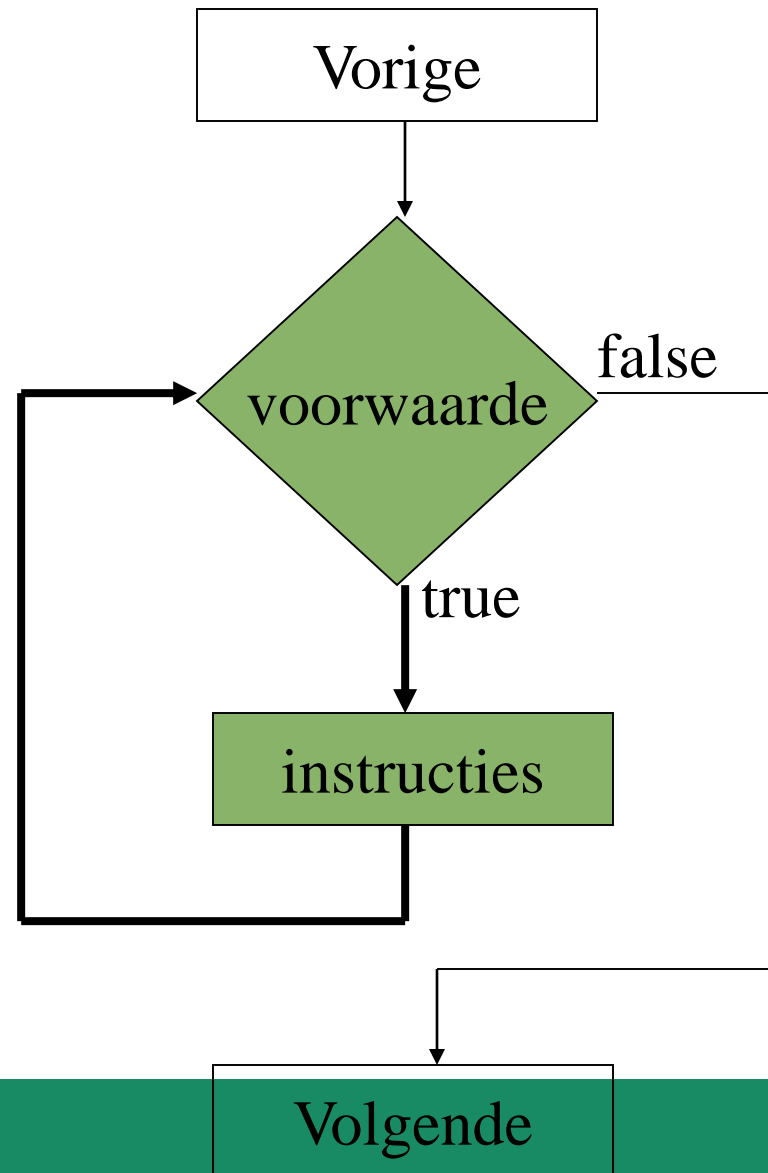
Vorige;

```
while (voorwaarde) {  
    //instructies  
}
```

Volgende;

→ De voorwaarde moet bepalend zijn

→ Instructies worden mogelijks niet uitgevoerd



Blijf in de lus zolang het regent ...

```
boolean regen;  
  
Scanner scan = new Scanner(System.in);  
System.out.println("Regent het?");  
regen = scan.nextBoolean();  
  
while (regen) {  
    System.out.println("Regent het?");  
    regen = scan.nextBoolean();  
}
```

Opgelet met lussen die blijven lopen ... verzorg je stopconditie !

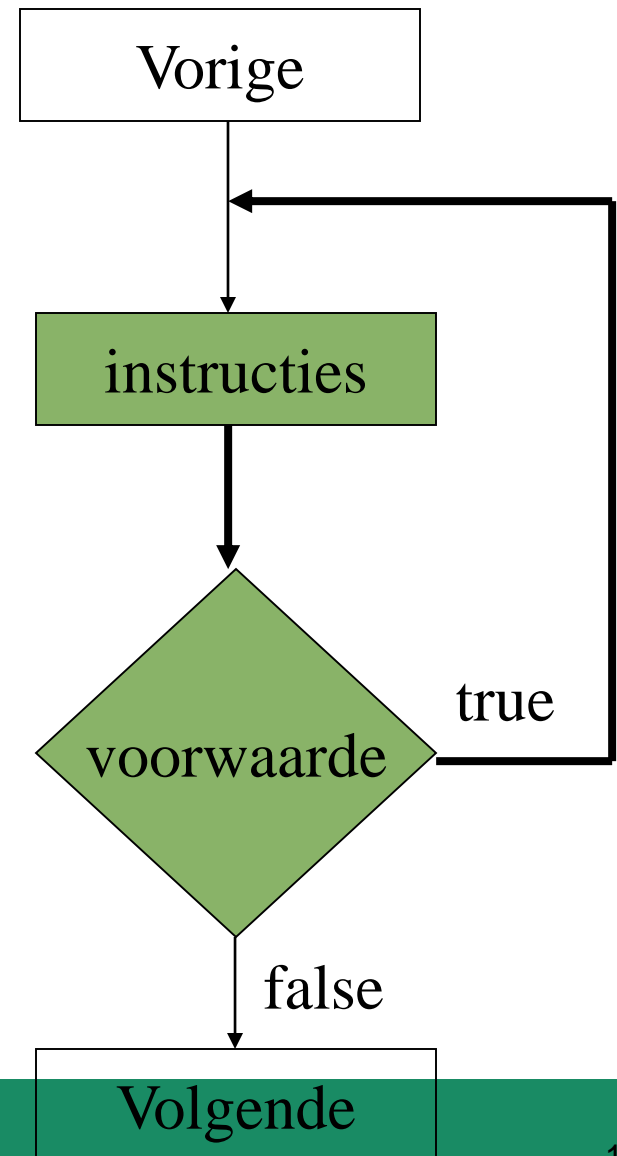
```
boolean gonzo;  
  
System.out.println("Gonzo did you  
    manage to play the trumpet ?");  
gonzo = false;  
  
while (!gonzo) {  
    System.out.println("Try again !!");  
    System.out.println("Gonzo did you  
        manage to play the trumpet ?");  
    gonzo = false;  
}
```



Iteraties (do-while)

```
Vorige;  
  
do {  
    //instructies  
} while (voorwaarde);  
  
Volgende;
```

- De voorwaarde moet bepalend zijn
- Instructies worden minimaal 1x uitgevoerd



Blijf in de lus zolang het regent ...

```
boolean regen;  
  
Scanner scan = new Scanner(System.in);  
  
do {  
    System.out.println("Regent het?");  
    regen = scan.nextBoolean();  
} while (regen)
```

1 gok

```
public class Gok {  
    public static void main(String[] args)  
        throws java.io.IOException {  
        char gok, oplossing = 'M';  
        System.out.println("Gok de juiste letter tussen A  
                            en Z : ");  
        gok = (char) System.in.read();  
        if (gok == oplossing)  
            System.out.println("Proficiat !! Juist gegokt ! ");  
        else {  
            System.out.print("... Jammer, je gok was ");  
            // geneste if  
            if (gok < oplossing)  
                System.out.println("te laag !");  
            else  
                System.out.println("te hoog");  
        }  
    }  
}
```


-> meerdere gokken

```
public class Gok2 {  
    public static void main(String[] args) throws java.io.IOException {  
        char gok, oplossing = 'M', rommel;  
  
        do{  
            System.out.println("Gok de juiste letter tussen A en Z : ");  
            gok = (char) System.in.read();  
  
            do{  
                rommel = (char)System.in.read();  
            } while (rommel != '\n');  
  
            if (gok == oplossing)  
                System.out.println("Proficiat !! Juist gegokt ! ");  
            else {  
                System.out.print("... Jammer, je gok was ");  
                // geneste if  
                if (gok < oplossing)  
                    System.out.println("te laag !");  
                else  
                    System.out.println("te hoog");  
            }  
        } while(gok != oplossing);  
    }  
}
```

vb. Geef een geldige maand in

```
int maand;  
  
do {  
    System.out.println("Geef een maand in (1..12)");  
    maand = Input.readInt();  
} while (maand < 1 || maand > 12);
```

while

versus

do-while

```
int getal;  
  
while (getal <= 100) {  
    getal = ...; //vraag getal  
}
```

variable might not
have been initialized

```
int getal = 123;  
  
while (getal <= 100) {  
    getal = ...; //vraag getal  
}
```

Lus wordt nooit uitgevoerd

```
do {  
    int getal = ...; //vraag getal  
} while (getal <= 100);
```

Cannot find symbol
Symbol : variable getal

```
int getal = 0;  
  
while (getal <= 100) {  
    getal = ...; //vraag getal  
}
```

```
int getal;  
  
do {  
    getal = ...; //vraag getal  
} while (getal <= 100);
```

vb. Meet de lengte van je tuin met een rolmeter van 5m lengte

Zolang je niet op het einde van je tuin bent gekomen,
meet een volgend stukje en tel op bij de reeds gemeten afstand!

Wat is de uitvoer ?

```
int lengteTuin = 0;
boolean bots = true; //true

while (!bots) {
    lengteTuin += 5;
}

System.out.println(lengteTuin);
```

Wat is de uitvoer ?

```
int lengteTuin = 0;
boolean bots = false; //false

while (!bots) {
    lengteTuin += 5;
}

System.out.println(lengteTuin);
```

```

public class Tuin {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int rolmeter = 5; //in meter
        double lengteTuin = 0; //in meter

        System.out.println("Raak je de afsluiting al (j/n) ?");
        char antwoord = scan.next().charAt(0);

        while (antwoord == 'n' || antwoord == 'N') {
            lengteTuin += rolmeter;
            System.out.println("Raak je de afsluiting al (j/n) ?");
            antwoord = scan.next().charAt(0);
        }

        System.out.println("LeesMeterAf (in cm):");
        lengteTuin += scan.nextInt() / 100.0;

        System.out.println("Je tuin is " + lengteTuin + " meter lang");
        scan.close();
    }
}

```

```

public class Tuin {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int rolmeter = 5; //in meter
        double lengteTuin = 0; //in meter

        System.out.println("Raak je de afsluiting al (j/n) ?");
        char antwoord = scan.next().charAt(0);

        while (antwoord != 'j' && antwoord != 'J') {
            lengteTuin += rolmeter;
            System.out.println("Raak je de afsluiting al (j/n) ?");
            antwoord = scan.next().charAt(0);
        }

        System.out.println("LeesMeterAf (in cm):");
        lengteTuin += scan.nextInt() / 100.0;

        System.out.println("Je tuin is " +lengteTuin+ " meter lang");
        scan.close();

    }
}

```

Vraag: Wat gebeurt er nu als je '?' ingeeft ?

```

public class Tuin {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int rolmeter = 5; //in meter
        double lengteTuin = 0; //in meter

        System.out.println("Raak je de afsluiting al (j/n) ?");
        char antwoord = scan.next().charAt(0);

        while (antwoord != 'j' && antwoord != 'J') {
            if (antwoord == 'n' || antwoord == 'N') {
                lengteTuin += rolmeter;
            }

            System.out.println("Raak je de afluiting al (j/n) ?");
            antwoord = scan.next().charAt(0);
        }

        System.out.println("LeesMeterAf (in cm):");
        lengteTuin += scan.nextInt() / 100.0;

        System.out.println("Je tuin is " + lengteTuin + " meter lang");
        scan.close();
    }
}

```

```

public class Tuin {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int rolmeter = 5; //in meter
        double lengteTuin = 0; //in meter

        char antwoord;

        do {
            System.out.println("Raak je de afsluiting al (j/n) ?");
            antwoord = scan.next().charAt(0);

            if (antwoord == 'n' || antwoord == 'N') {
                lengteTuin += rolmeter;
            }
        } while (antwoord != 'j' && antwoord != 'J');

        System.out.println("LeesMeterAf (in cm):");
        lengteTuin += scan.nextInt() / 100.0
        System.out.println("Je tuin is " +lengteTuin+ " meter lang");
        scan.close();
    }
}

```



```
public class Tuin {  
    public static void main(String[] args) {  
        int rolmeter = 5; //in meter  
        double lengteTuin = 0; //in meter
```

```
        while (true) {  
            System.out.println("Raak je de afsluiting al (j/n) ?");  
            char antwoord = scan.next().charAt(0);  
  
            if (antwoord == 'n' || antwoord == 'N') {  
                lengteTuin += rolmeter;  
            }  
            else if (antwoord == 'j' || antwoord == 'J') {  
                break;  
            }  
            else {  
            }  
        }  
    }
```

```
        System.out.println("LeesMeterAf (in cm):");  
        lengteTuin += scan.nextInt() / 100.0  
        System.out.println("Je tuin is " +lengteTuin+ " meter lang");  
        scan.close();  
    }
```

```
}
```

Algemene vorm van een `while`-lus:

```
while (booleaanse uitdrukking) {  
    ...  
}
```

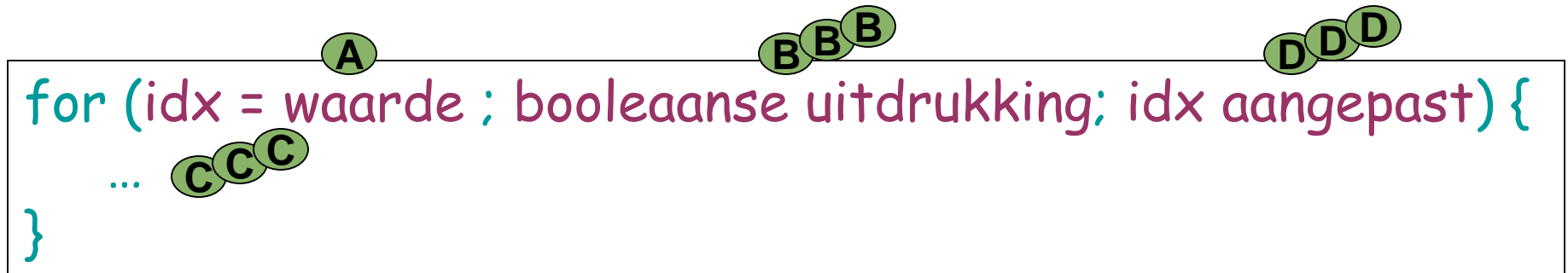
Algemene vorm van een `do/while`-lus:

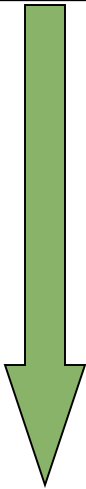
```
do {  
    ...  
} while (booleaanse uitdrukking);
```

Algemene vorm van een `for`-lus:

```
for (idx = waarde ; booleaanse uitdrukking ; idx aangepast) {  
    ...  
}
```

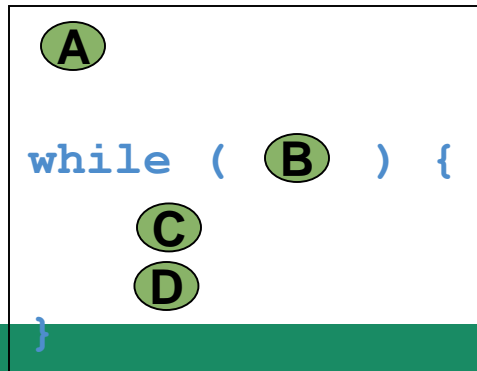
In elke for-lus 'herken' je een while-lus...

A diagram of a for loop: `for (idx = waarde ; booleaanse uitdrukking; idx aangepast) { ... }`. The labels are: A above the opening parenthesis, B above the semicolon after the condition, C above the semicolon after the increment, and D above the closing parenthesis.



`for (char c = 'A' ; c <= 'Z' ; c++) {
 System.out.println(c);
}`



A diagram of a while loop: `while (B) { ... }`. The labels are: A above the opening parenthesis, B above the condition, C above the opening curly brace, and D above the closing curly brace.

`char c = 'A';

while (c <= 'Z') {
 System.out.println(c);
 c++;
}`

Wanneer welke loop gebruiken?

- Gebruik de **for-lus** wanneer je vooraf weet hoe vaak je zal moeten herhalen
- Gebruik **do-while** wanneer je minstens 1x het codeblok moet uitvoeren
- Gebruik **while** wanneer je vooraf helemaal niet weet of en hoelang je zal moeten herhalen