

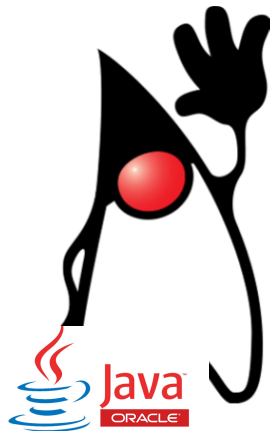
# Controle Structuren in Java

Klasgroep	1EO-ICT
Opleiding	Bachelor Elektronica-ICT
Lokaal	Groot Auditorium
Tijdstip	maandag lestijd 3
Docent	Katja Verbeeck
Contact	<a href="mailto:katja.verbeeck@odisee.be">katja.verbeeck@odisee.be</a>
Handboek	hfst 3 - p 64 - p74



# Inhoud

- 1 Introductie
- 2 Keuze structuren
  - enkelvoudige if
  - nested if
- 3 Meervoudige if
  - if-else-if
  - switch
  - Oefeningen

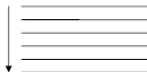


# De control flow

Het programma verloop (control flow) tot nogtoe is simpel : alle instructies binnen de main methode worden **sequentieel** uitgevoerd. Java voorziet echter control structuren om bepaalde blokken code over te slaan, of om bepaalde blokken code meermaals uit te voeren.

# De control flow

Binnen een methode worden instructies normaal sequentieel uitgevoerd:



Nieuw: een keuze maken  
uit alternatieve stukken code



Nieuw: een stuk code  
een aantal keer uitvoeren



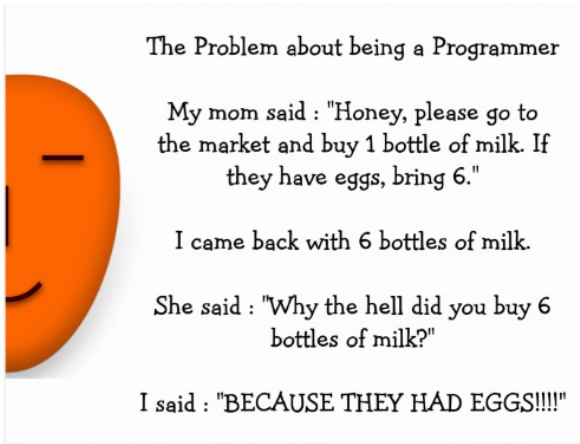
## De selecties:

- **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**

# Keuzes maken



# Een keuze maken op basis van een booleaanse expressie

if you find a moose ...

... you can make the chocolate moose



## in code ...

```
// vertaald naar java code
boolean moose;
...
if (moose == true) {
    System.out.println("Start making
        chocolate mousse !!! ");
    System.out.println("Rub the moose with
        the chocolate ... ");
}
```

```
// of nog korter
boolean moose;
...
if (moose) {
    System.out.println("Start making
        chocolate mousse !!! ");
    System.out.println("Rub the moose with
        the chocolate ... ");
}
```



# en met een alternatief

if you find a moose ...

... you can make the chocolate moose

or else ...

... go and find your moose first !!!

## en met een alternatief

```
boolean moose;  
...  
if (moose) {  
    System.out.println("Start making  
        chocolate mousse !!! ");  
    System.out.println("Rub the moose with  
        the chocolate ... ");  
}  
else{  
    System.out.println("Go and find your  
        moose first !!! ");  
}
```

# if vs ternaire operator

“Can I copy your HW?”

“Yeah just change it up a bit so it doesn't look obvious you copied.”

```
if(count > 10){  
    total += 15;  
}  
else{  
    total += 0;  
}
```

```
total = (count > 10) ? (total + 15) : (total + 0);
```

# nested-ifs

De java expressie binnen de **if** of **else** tak mag opnieuw een if-structuur zijn, en de java expressie binnen deze laatste if of else tak mag opnieuw een if-structuur zijn, en de java expressie binnen deze laatste if of else tak mag opnieuw een if-structuur zijn, en de ....

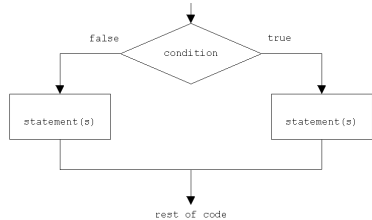
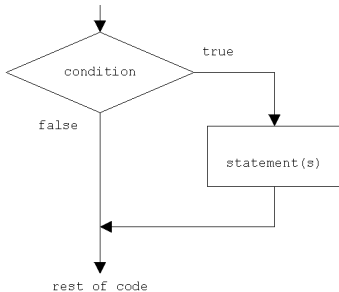
```

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.printl("te hoog");
        }
    }
}

```

# Algemeen : Enkelvoudige if



# Meer dan 1 keuze ... en meer dan 1 actie

if you find a chicken ...  
and if you find eggs ...  
or pingpong balls ...

... you can boil some eggs ... or you can eat the chicken

## in code ...

```
boolean chicken, hasEggs, hasPingPongBalls;

if (chicken && hasEggs)
    System.out.println("Enjoy your Eggs !! ");
else if (chicken && hasPingPongBalls){
    System.out.println("you cannot eat
        pingpongballs ?!!");
    System.out.println(" You better eat the
        chicken !! ");
}
else if (!chicken)
    System.out.println("I am sorry no chicken - no
        eggs");
```



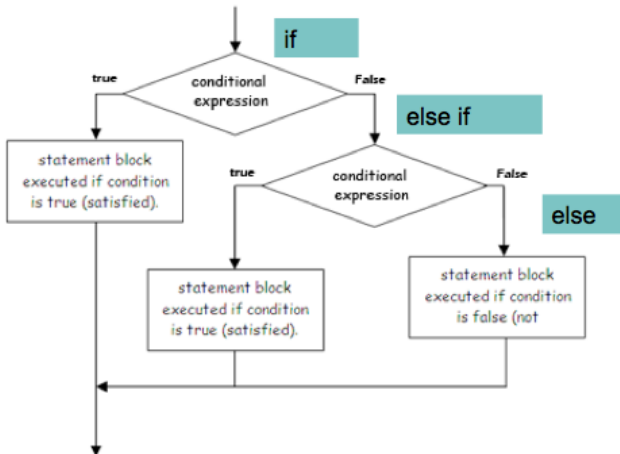
# De if-else-if ladder

```
import java.util.Scanner;

public class Ladder {
    public static void main(String[] args){
        System.out.println("Uit welk werelddeel kom je? ");
        Scanner scan = new Scanner(System.in);
        String deel = scan.next();

        if (deel.equals("Europa"))
            System.out.println("Europa? Dan ben je Europeaan");
        else if (deel.equals("Afrika"))
            System.out.println("Afrika? Dan ben je Afrikaan");
        else if (deel.equals("Noord-Amerika"))
            System.out.println("Noord Amerika ? Dan ben je Noord Amerikaan");
        else if (deel.equals("Zuid-Amerika"))
            System.out.println("Zuid Amerika? Dan ben je Zuid Amerikaan");
        else if (deel.equals("Azie"))
            System.out.println("Azi\u00EB? Dan ben je Aziaat");
        else if (deel.equals("Oceanie"))
            System.out.println("Oceani\u00EB? Dan ben je Oceani\u00EBr");
        else
            System.out.println("dan ben je waarschijnlijk een alien ... ");
    }
}
```

# Algemeen: meervoudige keuze



# Algemeen: meervoudige keuze

**SAMENVATTING:** Meest uitgebreide vorm van een if-structuur:

```
if (booleaanse uitdrukking) {  
    ...  
}  
else if (booleaanse uitdrukking) {  
    ...  
}  
...  
else {  
    ...  
}
```

# Meer dan 1 keuze ... voor elke waarde 1

which dish will he prepare?

- ... chocolate mousse ? — > you' d better find a moose
- ... doughnuts ? — > you need a riffle to shoot a hole in the muffin !
- ... frogg legs? — > catch Kermit or Robin
- ... chicken in the basket ? — > throw the chicken through a basket ball hoop

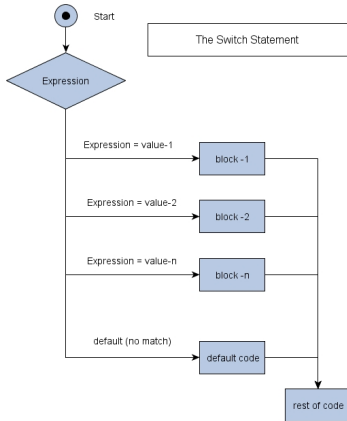
## in code ...

```
switch(dish){  
    case "chocolate mousse" :  
        System.out.println("you' d better find a moose ");  
        break;  
    case "doughnuts" :  
        System.out.println("you need a riffle to shoot a hole  
            in the muffin !");  
        break;  
    case "frogg legs" :  
        System.out.println("catch Kermit or Robin");  
        break;  
    case "chicken in the basket":  
        System.out.println("throw the chicken through a  
            basket ball hoop");  
        break;  
    default:  
        System.out.println("We're sorry, the chef won't  
            prepare this ! ");  
}
```

# break?

```
public class NoBreak {  
    public static void main(String[] args){  
        int i = 3;  
        switch(i) {  
            case 0:  
                System.out.println("Zero ");  
            case 1:  
                System.out.println("One ");  
            case 2:  
                System.out.println("Two ");  
            case 3:  
                System.out.println("Three ");  
            case 4:  
                System.out.println("Four ");  
            default:  
                System.out.println("Boom ");  
        }  
    }  
}
```

# Algemeen: meervoudige keuze



De expressie waarop je switch kan 1 van de volgende types zijn

- byte
- short
- int
- char
- String
- enumeraties (zie later)

Alle cases die voldaan zijn worden uitgevoerd tenzij een break statement de control flow uit de switch haalt. Cases kunnen gegroepeerd worden. Wanneer geen case voldoet wordt de default case uitgevoerd. De *default* case is optioneel.

# Algemeen: switch

Vorige blok code;

```
switch (var) {  
    case waarde1:  
        //instructies1  
        break;  
    case waarde2:  
        //instructies2  
        break;  
    case waarde3:  
        //instructies3  
        break;  
    ...  
    default:  
        //instructies  
        break;  
}
```

Volgende blok code;



# Oef

Schrijf code om een score in te lezen en geef een indicatie of de gebruiker al dan niet geslaagd was voor een test ( $\geq 5$  is geslaagd) . Indien geslaagd, druk af of het goed (7 of 8) of zeer goed was (9 of 10).



```
import java.util.Scanner;
public class Cijfer {
    public static void main(String[] args){
        System.out.println("Geef je score : ");
        Scanner scan = new Scanner(System.in);
        int cijfer = scan.nextInt();
        String info = "";

        if (cijfer >= 5){
            info += "Proficiat ! Geslaagd";
            if (cijfer > 8)
                info += ", zeer goed !";
            else if (cijfer > 6)
                info += ", goed !";
        }
        else
            info += "Jammer, niet geslaagd.";
        System.out.println(info);
    }
}
```

```
switch(cijfer){  
    // groepeer de cases  
    case 0:  
    case 1:  
    case 2:  
    case 3:  
    case 4: info += "Jammer, niet geslaagd.";  
            break;  
    case 5:  
    case 6: info += "Proficiat ! Geslaagd";  
            break;  
    case 7:  
    case 8: info += "Proficiat ! Geslaagd, goed  
                !";  
            break;  
    case 9:  
    case 10: info += "Proficiat ! Geslaagd, zeer  
                goed !";  
            break;  
}
```

```
switch(cijfer){  
    // groepeer de cases  
    case 0:  
    case 1:  
    case 2:  
    case 3:  
    case 4: info += "Jammer, niet geslaagd.";  
            break;  
    case 5:  
    case 6:  
    case 7:  
    case 8:  
    case 9:  
    case 10: info += "Proficiat ! Geslaagd";  
    default: if (cijfer > 8)  
                info += ", zeer goed!";  
            else if (cijfer > 6)  
                info += " , goed!";  
}
```

# Oef

Schrijf een programma waarbij de gebruiker een maand ingeeft (via een cijfer 1 - 12) en bepaalt in welk kwartaal van het jaar deze maand valt. Doe dit op 3 verschillende manieren repectievelijk door gebruik te maken van

- een if-structuur
- een switch structuur
- een ternaire operator

```
import java.util.Scanner;
public class Kwartaal {
    public static void main(String[] args){
        System.out.println("Geef een maand in (1-12):");
        Scanner scan = new Scanner(System.in);
        int maand = scan.nextInt();
        String info = "";

        if(maand == 1 || maand == 2 || maand == 3)
            info += "Eerste kwartaal";
        else if(maand == 4 || maand == 5 || maand == 6)
            info += "Tweede kwartaal";
        else if(maand == 7 || maand == 8 || maand == 9)
            info += "Derde kwartaal";
        else if(maand == 10 || maand == 11 || maand ==
            12)
            info += "Vierde kwartaal";
        else info += "Dit is een ongeldige maand";
        System.out.println(info);
    }
}
```

```
import java.util.Scanner;
public class Kwartaal1 {
    public static void main(String[] args){
        System.out.println("Geef een maand in (1-12):");
        Scanner scan = new Scanner(System.in);
        int maand = scan.nextInt();
        int kwartaal = (maand - 1) / 3;
        String info = "";
        if(kwartaal == 0)
            info += "Eerste kwartaal";
        else if(kwartaal == 1)
            info += "Tweede kwartaal";
        else if(kwartaal == 2)
            info += "Derde kwartaal";
        else if(kwartaal == 3)
            info += "Vierde kwartaal";
        else info += "Dit is een ongeldige maand";
        System.out.println(info);
    }
}
```

## switch

```
switch(maand){  
    case 1:  
    case 2:  
    case 3: info += "Eerste kwartaal"; break;  
    case 4:  
    case 5:  
    case 6: info += "Tweede kwartaal"; break;  
    case 7:  
    case 8:  
    case 9: info += "Derde kwartaal"; break;  
    case 10:  
    case 11:  
    case 12: info += "Vierde kwartaal"; break;  
    default: info += "Dit is een ongeldige  
                maand";  
}
```



## ternaire operator

```
import java.util.Scanner;
public class Kwartaal3 {
    public static void main(String[] args){
        System.out.println("Geef een maand in (1-12):");
        Scanner scan = new Scanner(System.in);
        int maand = scan.nextInt();
        int kwartaal = (maand -1) / 3;
        System.out.println
            (kwartaal == 0 ? "Eerste kwartaal" :
             kwartaal == 1 ? "Tweede kwartaal " :
             kwartaal == 2 ? "Derde kwartaal " :
             kwartaal == 3 ? "Vierde kwartaal " :
             "Dit is een ongeldige maand");
    }
}
```