

2. if

H5. Beslissingen



Program flow

Program flow

Het pad dat een programma volgt (ook thread of execution genoemd)

Flow van eenvoudige programma's

- 3 basistypes:
 1. 'Straight line' (tot nu)
 2. Gekozen naargelang **een gegeven voorwaarde** (focus deze les)
 3. Herhaalt zolang een bepaalde voorwaarde geldt (loops)

Straight program flow

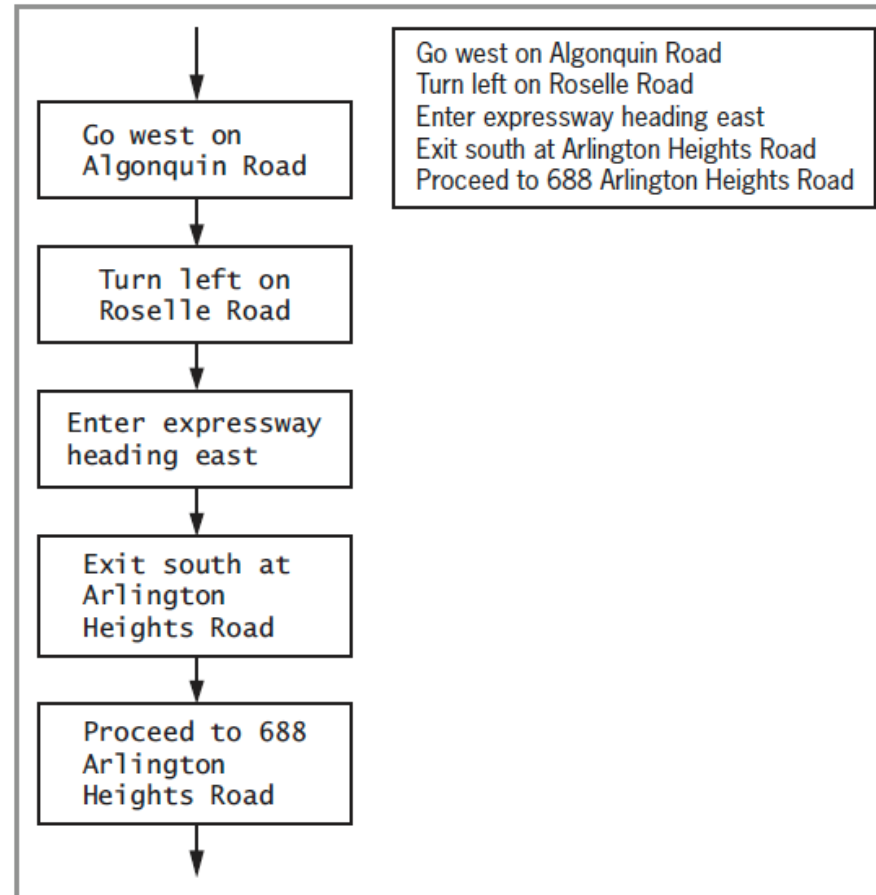


Figure 4-1 Flowchart and pseudocode for a series of sequential steps

Keuzes maken

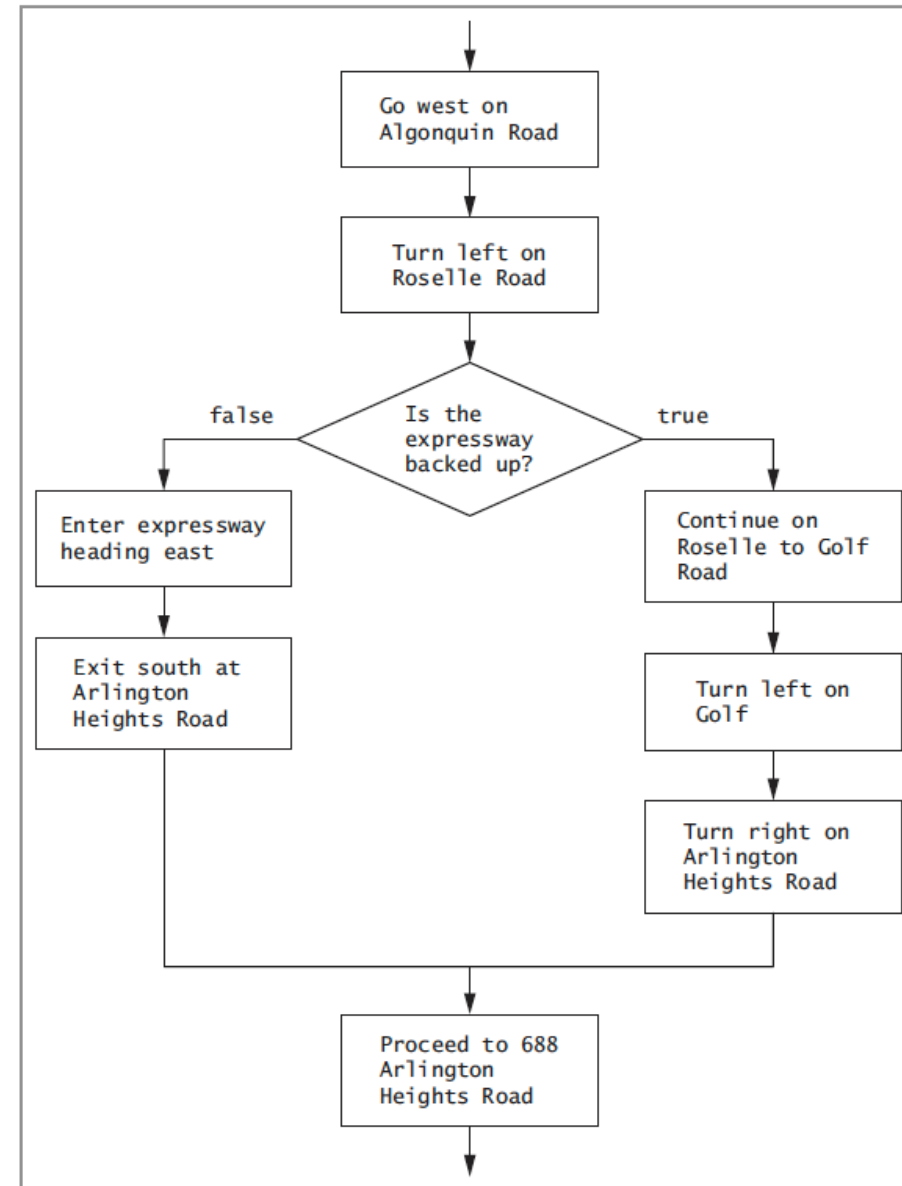
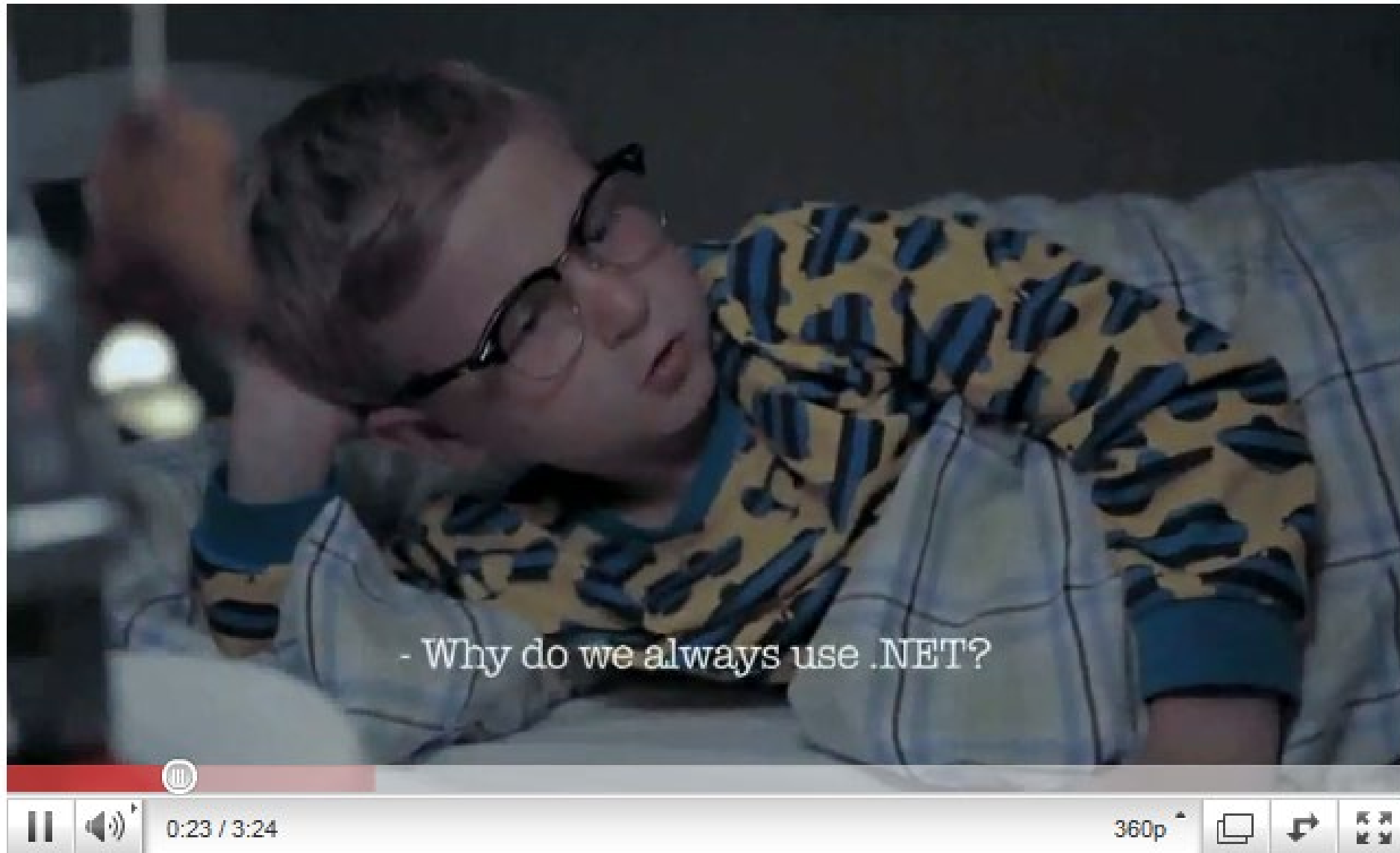


Figure 4-2 Flowchart including a decision



Definities



if statement

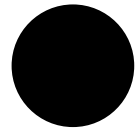
Gebruikt voor een enkelvoudig
alternatief pad aan te duiden
Afhankelijk van resultaat logische
expressie



Block { }

1 of meerdere statements
binnen een accolade paar

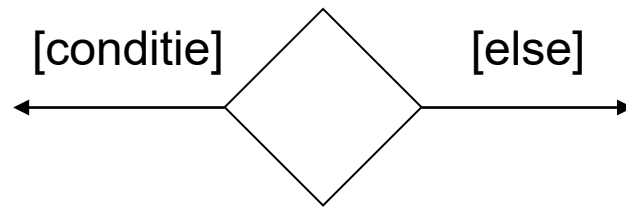
UML Activity diagram



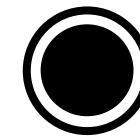
starttoestand



Actie, activiteit

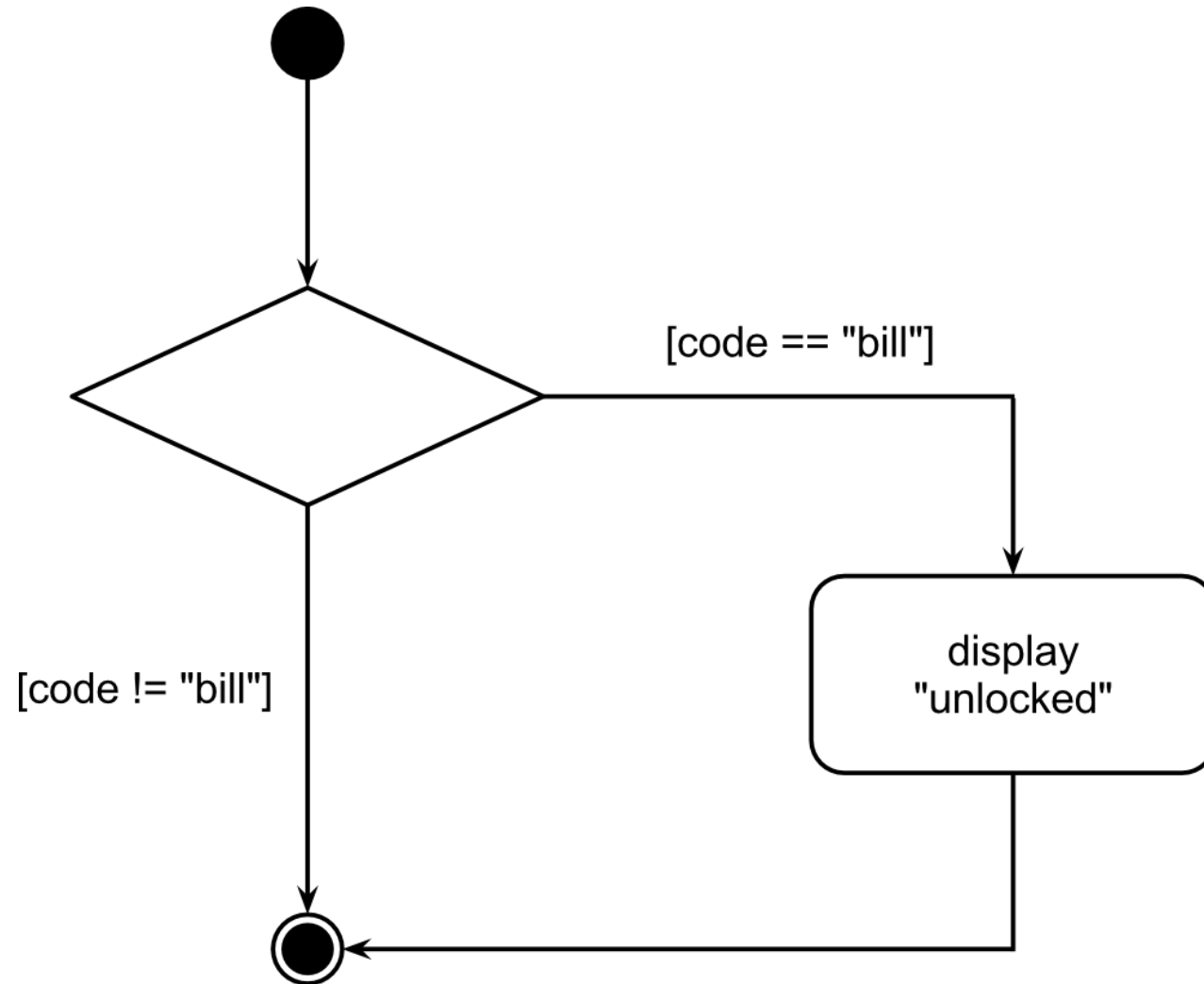


beslissing



eindtoestand

UML Activity diagram

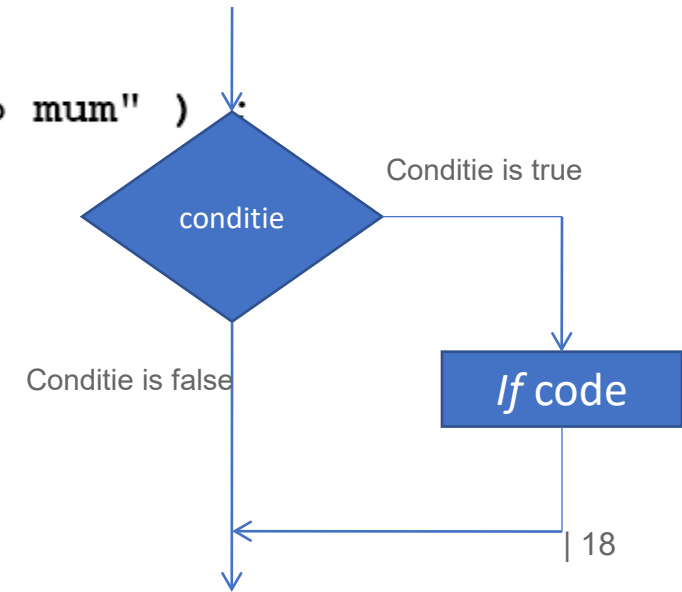


Condities

- Met bool kunnen we 'true' of 'false' status bepalen
- Een logische expressie heeft een bool als resultaat
- Voorbeeld

```
if (true)
{
    Console.WriteLine ( "hello mum" )
}
```

Redelijk domme conditie, daar deze altijd 'true' is en de code dus altijd uitgevoerd zal worden.

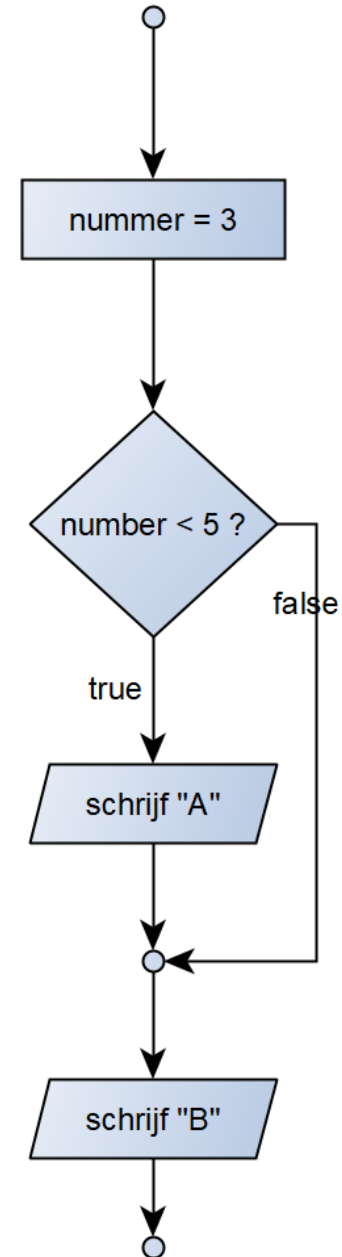


Voorbeeld if

```
int number = 3;  
  
if ( number < 5 )  
    Console.WriteLine ("A");  
Console.WriteLine("B");
```

- Uitvoer:

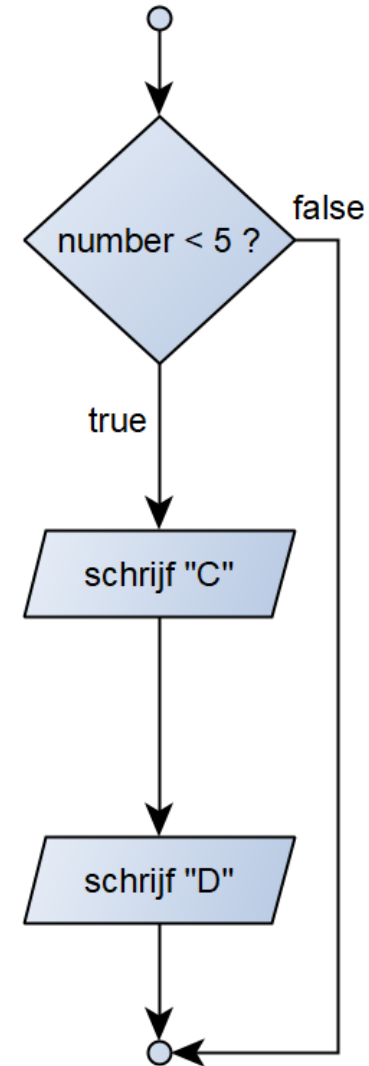
A
B



If met block

- Gebruik altijd block {} na if!

```
if ( number < 5 )  
{  
    Console.WriteLine ("C");  
    Console.WriteLine ("D");  
}
```



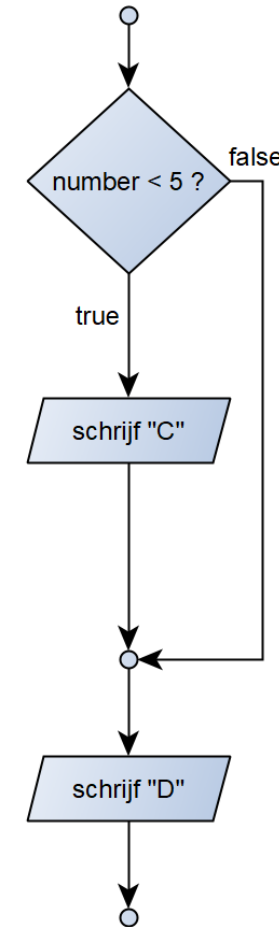
Veel gemaakte fouten



Veel gemaakte fout 1: accolades vergeten

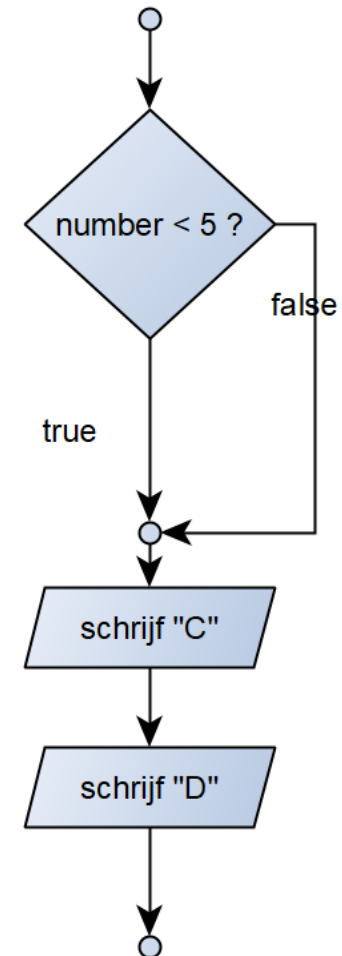


```
if ( number < 5 )  
    Console.WriteLine ("C");  
    Console.WriteLine ("D");
```



Veel gemaakte fout 2: puntkomma na if

```
if ( number < 5 );  
    Console.WriteLine ("C");  
    Console.WriteLine ("D");
```



If-else statement

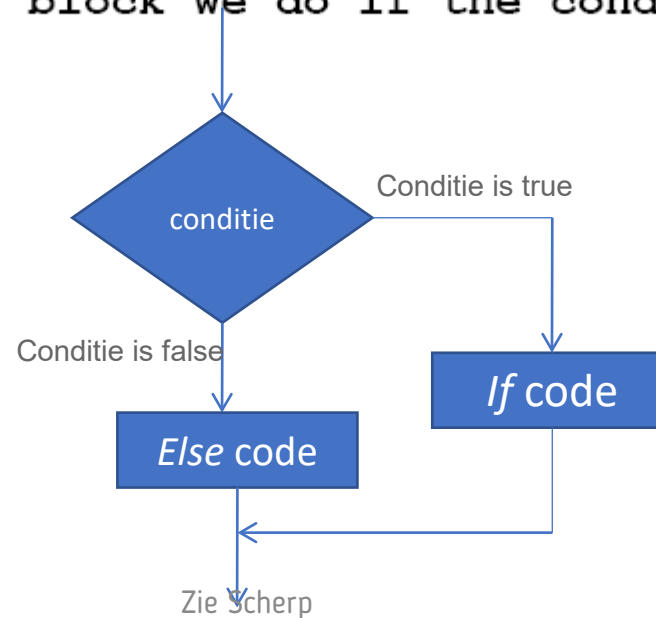
- **Dual-alternative decisions**
 - Heeft 2 mogelijke uitkomsten
- **if-else statement**
 - Gebruikt om een actie (of block acties) indien de expressie *true* is
 - Een andere, alternatieve actie(s) indien de expressie *false* is

Voorwaardelijke uitvoer: if/else

- Gebruiken om af te takken in programma

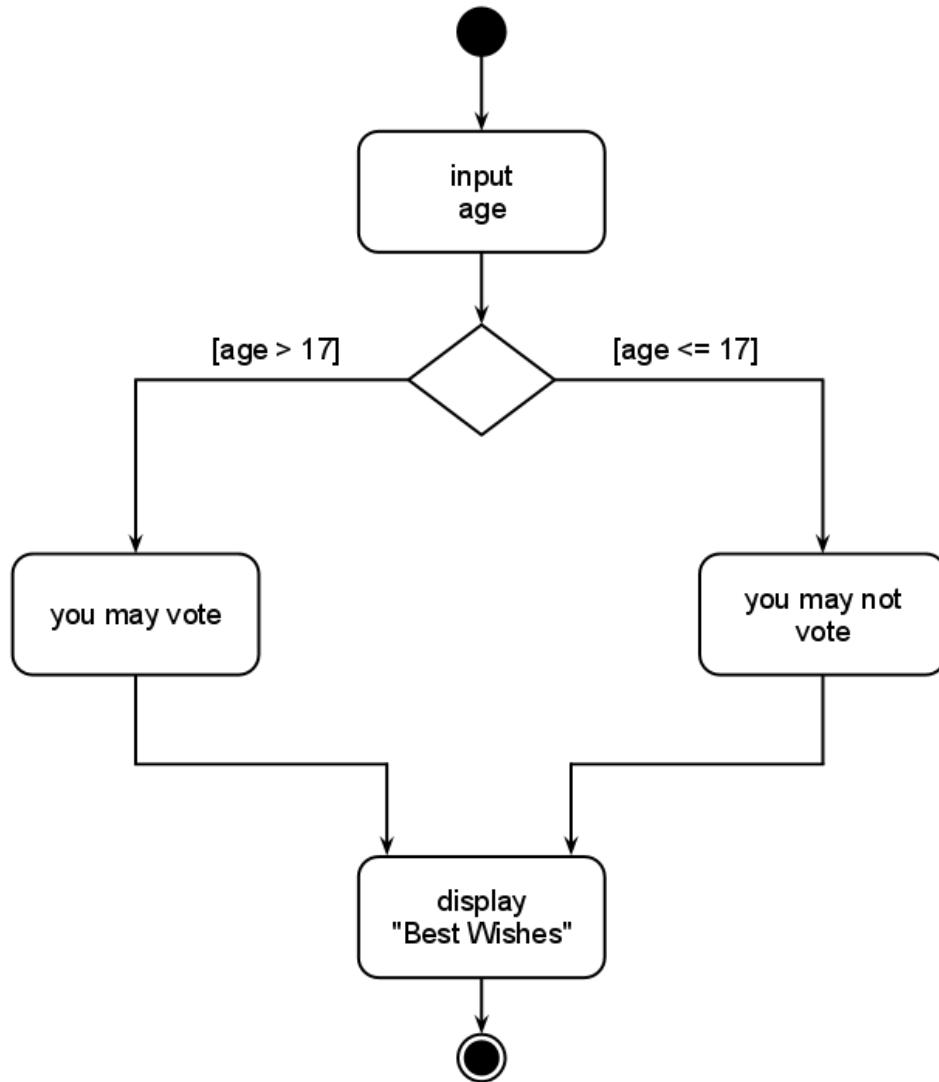
- Syntax:

```
if (condition)
    statement or block we do if the condition is true
else
    statement or block we do if the condition is false
```



If-else statement

```
int x = 10;  
  
if ( x > 9 )  
{  
    Console.WriteLine ("x is greater than 9!");  
}  
else  
{  
    Console.WriteLine ("x is less than 9!");  
}
```



```
int age;  
age = Convert.ToInt32(Console.ReadLine());  
  
if (age > 17)  
{  
    Console.WriteLine("You may vote.");  
}  
else  
{  
    Console.WriteLine("You may not vote");  
}  
Console.WriteLine("Best Wishes");
```

If/else if

```
int x = 9;

if (x == 10)
{
    Console.WriteLine ("x is 10");
}
else if (x == 9)
{
    Console.WriteLine ("x is 9");
}
else if (x == 8)
{
    Console.WriteLine ("x is 8");
}
```

Geneste if structuur

- **Nested if**
 - Een keuzestructuur bevindt zich in een andere keuze structuur

Geneste ifs

```
int reactorTemp = 1500;
string emergencyValve = " ";

if (reactorTemp < 1000)
{
    Console.WriteLine("Reactor temperature normal");
}
else
{
    Console.WriteLine("Reactor temperature too high!");
    if (emergencyValve == "closed")
    {
        Console.WriteLine("Reactor meltdown in progress!");
    }
}
```

Geneste ifs

```
if (age > 6)
{
    if (age < 16)
    {
        Console.WriteLine("junior rate.");
    }
    else
    {
        Console.WriteLine("adult rate.");
    }
}
else
{
    Console.WriteLine("child rate.");
}
```


Vereenvoudigen ifs

```
if ((age > 6) && (age < 16))
{
    Console.WriteLine ("junior rate.");
}
else if (age >= 16)
{
    Console.WriteLine ("adult rate.");
}
else
{
    Console.WriteLine ("child rate.");
}
```

GOOD CODERS...



... KNOW WHAT THEY'RE DOING

Programmer's Point

“Break down your conditions”

Probeer niet alles in één if te zetten, maar overwegen om op te splitsen in aparte if's onder mekaar.

Dus ipv:

```
if ((lives < 10 ) && ( gunpower > 100))  
    Console.WriteLine("Je sterft");
```

Overweeg:

```
if (lives < 10 )  
    if (gunpower > 100)  
        Console.WriteLine("Je sterft");
```

Lumping code together

- Alles tussen accolades { } wordt beschouwd als één blok en samen behandeld:

```
if ((lives < 10 )  && ( gunpower > 100))
{
    lives = 0;
    Console.WriteLine("Je bent doodgeschoten");
}
```

- Blokken in blokken mag ook, dit is **nesting**:

```
if ((lives < 10 )  && ( gunpower > 100))
{
    lives = 0;
    Console.WriteLine("Je bent doodgeschoten");

    if (extraLife == true)
    {
        lives = lives + 1;
        Console.WriteLine("Oef, je hebt nog een extra leven!");
    }
}
```

Accurate en correcte bereikcontroles uitvoeren

- **Range check**

- Een serie van if statement om te weten binnen welk bereik een bepaalde waarde valt

- Probleem: “comissie berekenen gebaseerd op verkochte producten”

```
if(saleAmount >= 1000)
    commissionRate = 0.08;
if(saleAmount >= 500)
    commissionRate = 0.06;
if(saleAmount <= 499)
    commissionRate = 0.05;
```

» DON'T DO IT

Although it was not the programmer's intention, both of the first two if statements are true for any saleAmount greater than or equal to 1000.

If - else if

- Oplossing

```
if(saleAmount >= 1000)
    commissionRate = 0.08;
else if(saleAmount >= 500)
    commissionRate = 0.06;
else commissionRate = 0.05;
```

Correct gebruik van && en || operator

- Probleem:
 - Toon een foutboodschap wanneer de payrate onder de 5.56 gaat **en** wanneer de payrate over de 60 gaat.
- Oplossing
 - ```
if (payRate < 5.65 || payRate > 60)
 Console.WriteLine ("Error in pay rate");
```

# Correct gebruik ! Operator (not operator)

- Probleem
  - Indien de salesCode niet gelijk is aan 'A' of 'B' krijgt de klant 10% korting.

- Oplossingen

```
if(salesCode != 'A' && salesCode != 'B')
 discount = 0.10;
```

```
if(!(salesCode == 'A' || salesCode == 'B'))
 discount = 0.10;
```