

SOFTWARE ONTWIKKELEN

AGENDA VOOR VANDAAG

- Lesdoelen
- Vorige les
- Branching
- Conditionele expressies
- Scope

LESDOELEN

Na de les van vandaag:

- Weet je wat een conditionele expressie is en kun je deze toepassen in C#.
- Pas je de if, else en else if toe in C#.
- Weet je wat het begrip scope inhoudt en pas je dit toe in C# code.

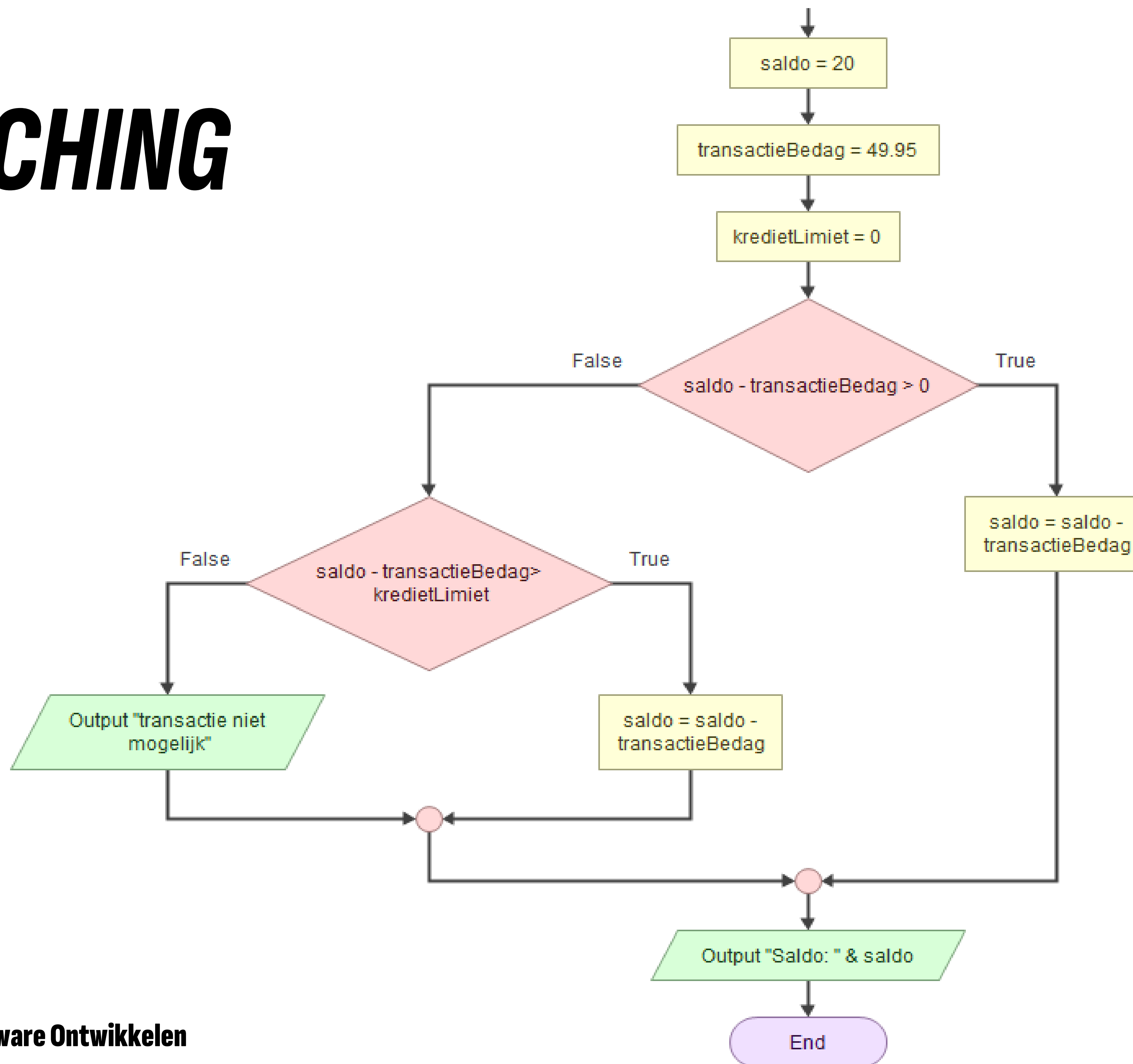
VORIGE LES

- Hoe gaat het met de planning?
- Schrijf op je wisbordje de volgende code:
 - Vraag om de naam van de gebruiker.
 - Toon in de console: "Welkom [ingevoerde naam]"

CODE REVIEW (10 MINUTEN)

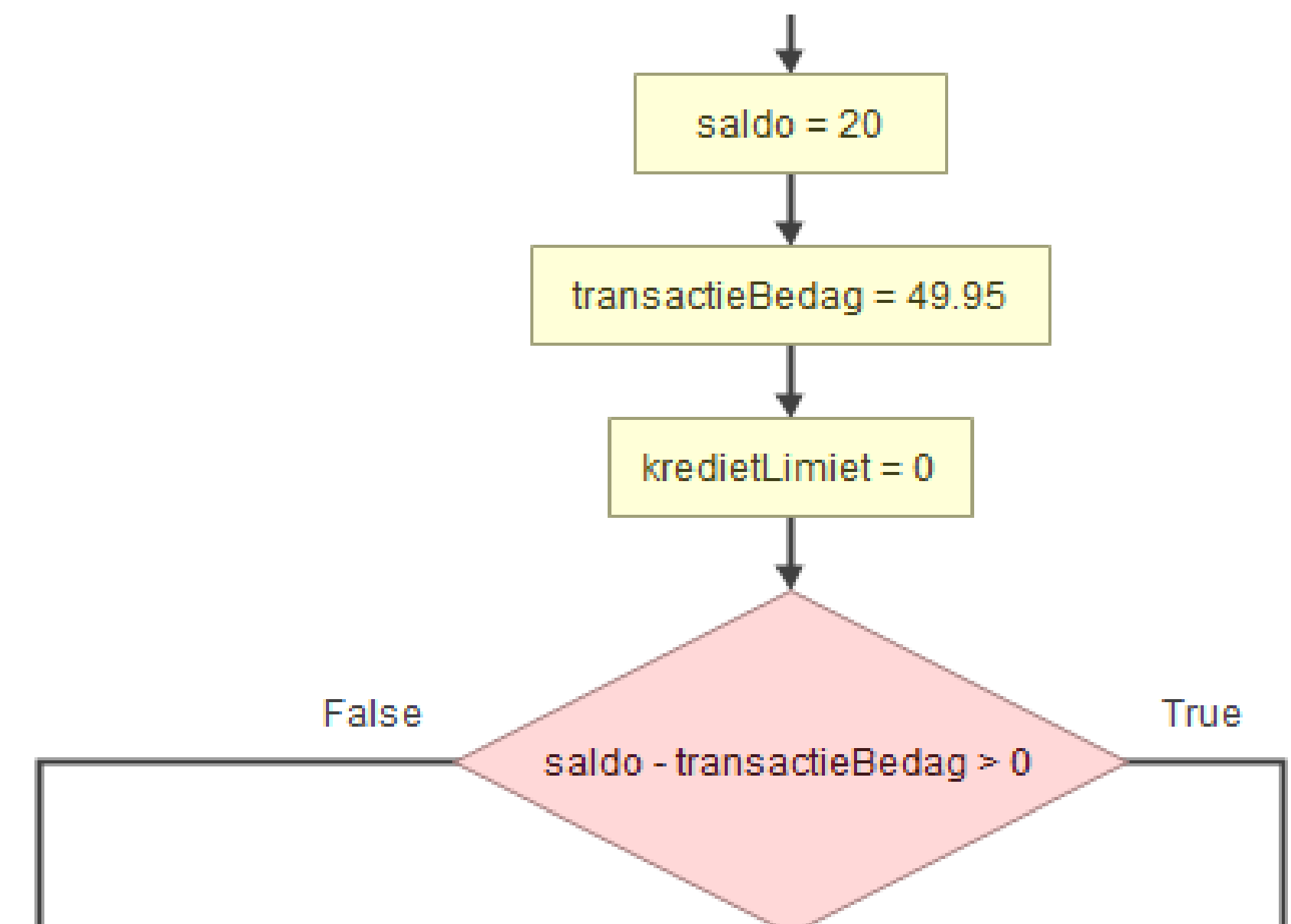
- Maak tweetallen en bekijk elkaars code (notebooks) van de opdrachten bij “Expressies”.
- Let hierbij vooral op:
 - Naamgeving variabelen.
 - Evt. fouten in de code!
- Bespreek je bevindingen met elkaar.

BRANCHING



BRANCHING

Programmeer mee.



CONDITIONELE EXPRESSIES

- Waar of niet waar -> true or false
- Gebruik je (o.a.) bij:
 - toekennen variabele type bool.
 - (else) if -> branching
 - while -> loops
- Hoeveel conditionele expressies in de code?
- Wat kan beter en schrijf een betere versie

```
double saldo = 419.33d;
double transactieBedrag = 49.95d;
bool saldoPositief = saldo >= 0;
double kredietLimiet = -1000.0d;

if (saldo - transactieBedrag >= 0)
{
    saldo -= transactieBedrag;
    Console.WriteLine("verwerk transactie");
}
else if (saldo - transactieBedrag > kredietLimiet)
{
    saldo -= transactieBedrag;
    Console.WriteLine("verwerk transactie");
    Console.WriteLine("Let op: saldo negatief");
}
else
{
    Console.WriteLine("blokkeer transactie");
}

Console.WriteLine($"Uw saldo is: {saldo}");
```


BEOORDEEL JE CODE!

- DRY
 - Worden alle variabelen gebruikt? Geen onnodige code?
 - Commentaar?
 - Nesting?
 - Samengestelde condities -> && ||
 - camelCase -> dus tweeWoorden voor variabelen.
 - PascalCase voor methodes en klassen.
-
- Is er een beste manier van programmeren?

```
decimal saldo = 20m;
decimal transactieBedrag = 49.90m;
decimal kredietLimiet = -1000.0m;

if (saldo - transactieBedrag >= 0 || saldo - transactieBedrag > kredietLimiet)
{
    saldo -= transactieBedrag;
    Console.WriteLine("verwerk transactie");
}
else
{
    Console.WriteLine("blokkeer transactie");
}

Console.WriteLine($"Uw saldo is: € {saldo:F2}");

if (saldo < 0) {
    Console.WriteLine("Let op: saldo negatief");
}
```

SCOPE

- Variabelen zijn slechts bekend in een 'omgeving'.
 - Logisch! Verdwenen zodra programma klaar.
- { } zorgt voor scope
- Hergebruik dus liever geen variabelenamen!

```
int a = 3;

if(a > 1)
{
    a += 5;
    int b = 6;
}

Console.WriteLine(a);
Console.WriteLine(b);
```

SCOPE

- Schrijf op je wisbordje:

Wat zijn de waarden van *a* en van *c* na uitvoeren van de code?

```
void DoeIets()  
{  
    int c = 15;  
}  
  
int a = 3;  
int c = 7;  
  
if (true)  
{  
    a += 5;  
    int b = 6;  
}  
  
DoeIets();  
Console.WriteLine(a);  
Console.WriteLine(c);
```




NU ZELF DOEN

- Schrijf een eenvoudig spelletje: 'raad het getal'. Je hebt 5 mogelijkheden om een getal onder de 10 te raden. Geef per keer aan: omhoog/ omlaag!
- Goed geraden: melding en stop programma.
- Hint:

```
var rand = new Random();  
int teRaden = rand.Next(1,11);  
Console.WriteLine(teRaden);
```

ASYNCHROON

- [Switch case - ATIx ICT-B1.1 Smart Meter Data Processing 2024-25 P1 \(avans.nl\)](#)
- [Enums - ATIx ICT-B1.1 Smart Meter Data Processing 2024-25 P1 \(avans.nl\)](#)