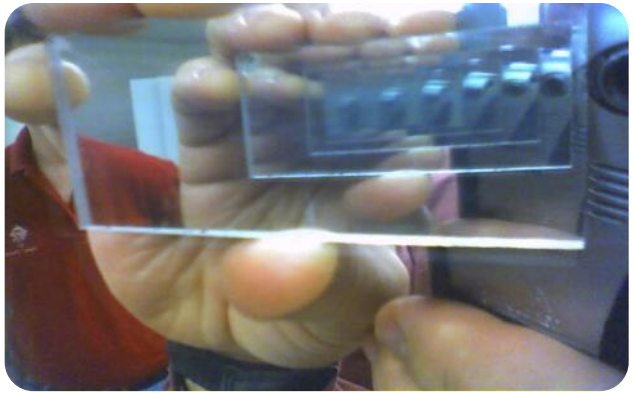
leren. durven. doen.

SYNTRA WEST

C# FUNDAMENTALS

C# PROGRAMMING

west-vlaanderen POM



Recursie

Methoden die zichzelf aanroepen

Inhoud

1. Wat is recursie?
2. Voorbeelden: Faculteit en CopyFolder
3. Recursie of Iteratie?

Wat is Recursie?

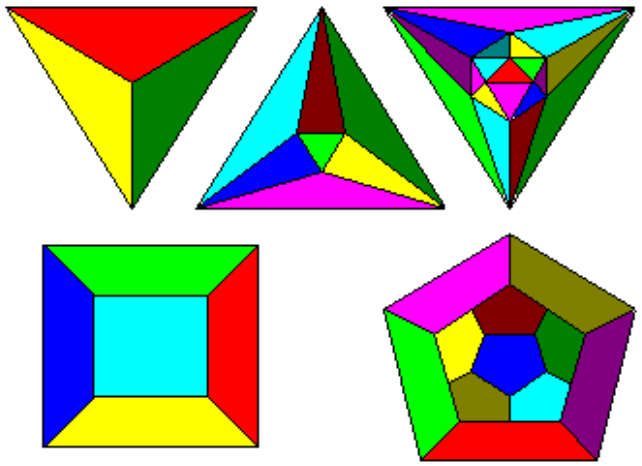
- Recursie betekent dat een methode zichzelf aanroept
- Recursie bevat
  - Direkte of indirekte recursieve method-aanroep
    - De methode roept zichzelf rechtstreeks aan of indirect via ander method(n)
  - Criteria om de recursie te eindigen
    - Voorkomt oneindige recursie

Recursie Faculteit – Voorbeeld

- Recursieve definitie van n! (n factulteit):

```
n! = n * (n-1)! for n >= 0
0! = 1
```

- 5! = 5 * 4! = 5 * 4 * 3 * 2 * 1 * 1 = 120
- 4! = 4 * 3! = 4 * 3 * 2 * 1 * 1 = 24
- 3! = 3 * 2! = 3 * 2 * 1 * 1 = 6
- 2! = 2 * 1! = 2 * 1 * 1 = 2
- 1! = 1 * 0! = 1 * 1 = 1
- 0! = 1



Recursie Faculteit – Voorbeeld

- Berekening van faculteit
  - 0! = 1
  - n! = n* (n-1)!, n>0

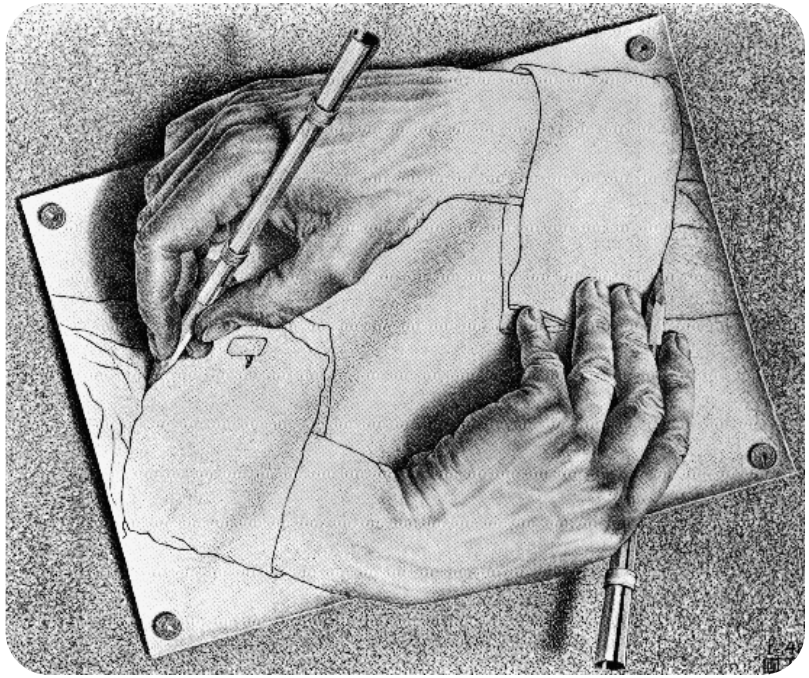
```
static void Main()
{
        Console.WriteLine(Faculteit(5));
}
static decimal Faculteit(decimal num)
{
        if (num == 0)
         return 1;
        else
         return num * Faculteit(num - 1);
}
```

Einde van de recursie

Recursie: de methode roept zichzelf aan

Recursieve methode Factulteit

Demo



Voorbeeld Recursieve CopyFolder

```
static void Main()
        {
            CopyFolder(@"C:\temptest", @"C:\tempcopy");
            Console.ReadLine();
        }
        static public void CopyFolder(string sourceFolder, string destFolder)
        {
            if (!Directory.Exists(destFolder))
                Directory.CreateDirectory(destFolder);
            string[] files = Directory.GetFiles(sourceFolder);
            foreach (string file in files)
            {
                string name = Path.GetFileName(file);
                string dest = Path.Combine(destFolder, name);
                File.Copy(file, dest);
            }
            string[] folders = Directory.GetDirectories(sourceFolder);
            foreach (string folder in folders)
            {
                string name = Path.GetFileName(folder);
                string dest = Path.Combine(destFolder, name);
                CopyFolder(folder, dest);
            }
        }
    }
```

Recursieve CopyFolder methode

Demo

Recursion or Iteration?

When to Use and When to Avoid Recursion?

Recursie kan CPU en/of geheugen inpalmen!

- Recursie kan veel cpu gebruiken vb:

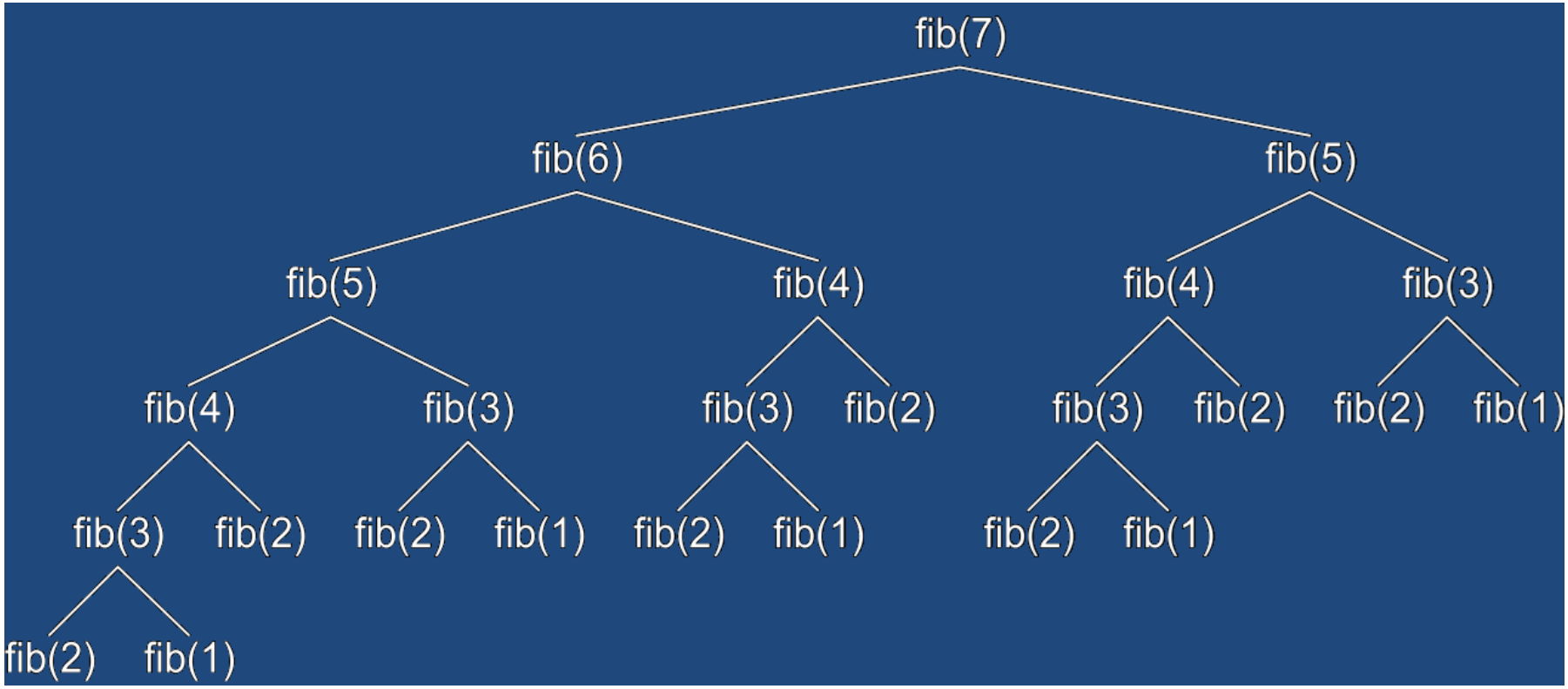
```
class Program
{
    static decimal Fibonacci(int n)
    {
        if ((n == 1) || (n == 2))
            return 1;
        else
            return Fibonacci(n - 1) + Fibonacci(n - 2);
    }

    static void Main()
    {
        Console.WriteLine(Fibonacci(10)); // 89
        Console.WriteLine(Fibonacci(50)); // Dit blijft hangen!
    }
}
```

Schadelijke recursie

Demo

Hoe werkt de recursieve Fibonacci?



- **fib(n) maakt ongeveer fib(n) recursieve aanroepen**
- **Dezelfde waarde wordt vele keren herberekend!**

Snelle Recursieve Fibonacci

- Elk berekend getal in de Fibonacci-reeks kan worden bijgehouden
- – En kan terug gegeven worden wanneer nodig bv.

```
static decimal[] fib = new decimal[55];
static decimal FibonacciSnel(int n)
{
    if (fib[n] == 0)
    {
        if ((n == 1) || (n == 2))
            fib[n] = 1;
        else
            fib[n] = FibonacciSnel(n - 1) + FibonacciSnel(n - 2);
    }
    return fib[n];
}
```

Snelle Recursieve Fibonacci

Demo

Wanneer Recursie gebruiken?

- Vermijd recursive wanneer een eenvoudig iteratief algoritme bestaat:
  - Voorbeelden: faculteit, Fibonacci reeks
- Gebruik recursieve algoritmen wanneer bij elke recursieve stap meer dan 1 mogelijkheid moet worden onderzocht:
  - Voorbeelden: permutaties, alle paden in labyrint
  - Indien er meer dan 1 recursieve aanroep is in de body van een recursieve method, kan dit iteratief worden aangeroepen (bv bij het berekenen van factulteit)

Samenvatting

- Recursie betekent dat een methode zichzelf aanroept
  - Het moet steeds een einde bevatten waar de recursieve aanroep stopt
- Krachtige techniek om combinatorische algoritmen te implementeren
  - Bv: generatie van permutaties, combinaties, variaties, maak ook voor bv 8 koninginnen-raadsel, schaakspel,…
- Recursie kan uw processor en/of geheugen compleet inpalmen wanneer niet correct gebruikt

Oefening recursieve methode

1. Schrijf een recursieve methode Power die de machtsverheffing berekent

```
public static void Main()
{
    Console.WriteLine("Geef een geheel getal:");
    int getal = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("Geef een gehele exponent:");
    int exponent = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("{0} tot de macht {1} is: {2}",getal,exponent,
        Power(getal, exponent));
}
```

leren. durven. doen.

SYNTRA WEST

Vragen?

west-vlaanderen POM