

Scope

EF IF

Aufgabe

- Welche Codezeilen führen zu Fehlern? Warum?

```
int a = 1;
{
    int a;      Fehler! a existiert schon
    int b = 7;
}
{
    int b;      Ist OK! Variable b gibt es zwar schon, aber in anderem Bereich
}
Console.WriteLine(a);
Console.WriteLine(b); Fehler! b existiert nicht in diesem Bereich
```

Scope

- de: Geltungsbereich, Sichtbarkeitsbereich
- Ist Codebereich, in dem eine Variable oder Funktion sichtbar ist, in dem sie so quasi 'lebt'

Scope

Variante A

```
1  int count = 1;
2
3  while (true)
4  {
5      Random rand = new Random();
6      int x = rand.Next(1, 10);
7      int y = rand.Next(1, 10);
8      int result = x + y;
9      Console.WriteLine("Calculation " + count + ": ");
10     Console.WriteLine(x + " + " + y + " = " + result + "\n");
11     count++;
12     Console.ReadKey();
13 }
14
```

VS.

Variante B

```
1  int count = 1;
2  Random rand = new Random();
3  int x, y, result;
4
5  while (true)
6  {
7      x = rand.Next(1, 10);
8      y = rand.Next(1, 10);
9      result = x + y;
10     Console.WriteLine("Calculation " + count + ": ");
11     Console.WriteLine(x + " + " + y + " = " + result + "\n");
12     count++;
13     Console.ReadKey();
14 }
```

- Kleine Gruppenarbeit:
 - Vergleiche die beiden Varianten.
 - Was sind Vorteile von A, Vorteile von B?
- Philosophie A: Variablen im mögl. eingeschränktem Scope
- Philosophie B: Möglichst wenige Deklarationen (-> effizienter)

Sinnvolle Variablennamen

```
1 int a;  
2 int b;  
3 int c;  
4 int d;  
5 int e;  
6 int f;
```

- Bereits so programmiert?
- Ausweg: Variablen umbenennen!
 - F2
 - oder rechte Maustaste und Umbenennen



Shortcuts & Trick

- Comment / Uncomment: CTRL + K + C / CTRL + K + U
- Format: CTRL + K + D
- Console.WriteLine(): cw + TAB
- Variablen umbenennen: F2 oder rechte Maustaste und Umbenennen