

# Vergleich zu C#

Florian Gehring

18.06.2020

# Hello, World!

```
1 using System;
2
3 namespace Beispielcode
4 {
5     class Program
6     {
7         static void Main(string[] args)
8         {
9             //Console.WriteLine("Hello World!");
10            Typesystem.types();
11        }
12    }
13 }
```

Dieses und viele folgende Beispiele: Tour of C#

- Von Microsoft Entwickelt
- „Direkter Konkurrent“ zu Java

- Stark Typisiert
  - jede Konstante, Variable und jeder Ausdruck hat einen Typ
- CTS (Common Type System)
  - **Jeder** Typ ist von `System.Object` (`object`) abgeleitet
- Integrierte Typen
  - „Zahlen“, `boolean`, `char`, `object`, `string`
- Referenztypen / Werttypen
  - `System.ValueType`
- Benutzerdefinierte Typen
  - `class`, `enum`, `struct`

# Variablen Deklarationen

```
1 using System;
2 using System.Linq;
3     // Declaration only:
4     float temperature;
5     string name;
6     Hello hello;
7
8
9     // Declaration with initializers (four examples):
10    char firstLetter = 'C';
11    var limit = 3;
12    int[] source = { 0, 1, 2, 3, 4, 5 };
13    var query = from item in source
14                  where item <= limit
```

- var Keyword
- List Comprehension

# Ganzzahlen

```
1  int i = 2;
2  // Get the Type
3  Type t = i.GetType();
4  Console.WriteLine(t);
5  // Which Methods can be called on this object?
6  foreach(var method in t.GetMethods()) {
7      Console.Write(method + " ");
8  }
9  Console.WriteLine();
10 // Print Parent classes.
11 Type b = t;
12 while(b != null) {
13     Console.Write(b + " -> ");
14     b = b.BaseType;
15 }
```

System.Int32

Int32.CompareTo(System.Object) Int32.CompareTo(Int32) ...

System.Int32 -> System.ValueType -> System.Object

## Java

- Primitive Typen nicht von Object abgeleitet
- Call-by-Value, Call-By-Reference
- Wrapper-Klasse Integer für int

## C#

- Alles (auch int) von object abgeleitet
- Zahlen, boolean sind „Werttype“
- int kann mit int? Nullable gemacht werden

# Codebeispiele 1