STRUKTUREN

Vincent Uhlmann
IT-Akademie Dr. Heuer GmbH

WAS SIND STRUKTUREN?

- Baupläne für Wertetypen
- Sie werden auf dem Stack gespeichert
- Strukturen sollten idealerweise unveränderlich (*immutable*) sein
- Sie sollten eine kleine Informationsmenge speichern (max. 16 Byte)
- Strukturen sind das Sprachmittel zur Definition von eigenen Wertetypen

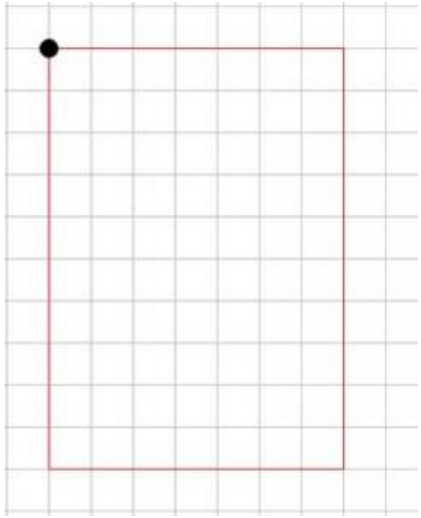
WAS SIND STRUKTUREN?

- Strukturen haben bei der Definition einige Einschränkungen
- Strukturen erben stets implizit von System.ValueType
- Strukturen können ihre Eigenschaften + Operationen nicht vererben
- Strukturen haben keinen Standard-Konstruktor
- Strukturen können ohne den new-Operator instanziiert werden

STRUKTUREN ERSTELLEN

- Beispiel:
  - Wir definieren ein Rechteck als Struktur
  - Ein Rechteck soll durch vier int-Werte im kartesischen Koordinatensystem bestimmt sein
  - Dazu nehmen wir die linke obere Ecke und die rechte untere Ecke

```
struct Rect
{
    public int TopLeftX;
    public int TopLeftY;
    public int BottomRightX;
    public int BottomRightY;
}
```



STRUKTUREN NUTZEN

- Strukturen können, wie andere Typen, instanziiert werden

```
Rect rect = new Rect();
rect.TopLeftX = 10;
rect.TopLeftY = 20;
rect.BottomRightX = 30;
rect.BottomRightY = 40;
```

```
Rect rect;
rect.TopLeftX = 10;
rect.TopLeftY = 20;
rect.BottomRightX = 30;
rect.BottomRightY = 40;
```

STRUKTUREN NUTZEN

- Bisher kann unsere Struktur nur Werte speichern, mehr nicht. Sie ist aktuell ein reiner Datenverbund!
- Strukturen können/sollen aber auch auf den Daten „arbeiten“ können, z.B. sinnvolle Berechnungen durchführen können.
- In unserem Fall eines Rechtecks, z.B. die Fläche und den Umfang berechnen können
- Dazu werden Methoden verwendet, die innerhalb der Struktur erstellt werden.
- Diese Methoden werden **ohne** das Schlüsselwort **static** erstellt, im Gegensatz zu den Methoden des vorherigen Moduls.

STRUKTUREN NUTZEN

```
struct Rect
{
    public int TopLeftX;
    public int TopLeftY;
    public int BottomRightX;
    public int BottomRightY;

    public int CalculateArea()
    {
        return (BottomRightX - TopLeftX) * (BottomRightY - TopLeftY);
    }
}
```

STRUKTUREN NUTZEN

- Die Funktionen, man nennt diese im Zusammenhang mit OOP-Code immer Methoden, findet auch wieder über den Member-Operator (.) statt:

```
Rect rect = new Rect();
rect.TopLeftX = 10;
rect.TopLeftY = 20;
rect.BottomRightX = 30;
rect.BottomRightY = 40;

Console.WriteLine(rect.CalculateArea());
```

DATETIME STRUKTUR

- Repräsentiert Datum und Uhrzeit
- Wird häufig für Zeitstempel, Kalenderdaten und Zeitzonen verwendet

- Eigenschaften
  - Now
  - UtcNow
  - …

```
DateTime now = DateTime.Now;
DateTime utc = DateTime.UtcNow;
Console.WriteLine(now);
Console.WriteLine(utc);
```

```
27.09.2024 09:05:30
27.09.2024 07:05:30
```

- Methoden
  - AddDays
  - AddHours
  - …

```
now = now.AddDays(5);
utc = utc.AddHours(15);
Console.WriteLine(now);
Console.WriteLine(utc);
```

```
02.10.2024 09:05:30
27.09.2024 22:05:30
```

TIMESPAN STRUKTUR

- Repräsentiert eine Zeitspanne (z.B. Dauer zwischen zwei Zeitpunkten (DateTime))
- Ideal für die Berechnung von Zeitdifferenzen

- Eigenschaften
  - Days
  - Hours
  - …
- Methoden
  - FromDays
  - Substract
  - …

```
TimeSpan timespan = TimeSpan.FromDays(5);
Console.WriteLine(timespan);
Console.WriteLine(timespan.Days);

DateTime dateTime1 = DateTime.Now;
DateTime dateTime2 = dateTime1.AddDays(5);
TimeSpan span = dateTime1.Subtract(dateTime2);
Console.WriteLine(dateTime1);
Console.WriteLine(dateTime2);
Console.WriteLine(span);
```

```
5.00:00:00
5

27.09.2024 09:15:56
02.10.2024 09:15:56
-5.00:00:00
```

GUID STRUKTUR

- Stellt eine eindeutige Identifikationsnummer (UUID) dar
- Nützlich, um eindeutige Schlüssel für Objekte zu erstellen

- Eigenschaften
  - Empty
  - …
- Methoden
  - NewGuid
  - Equals
  - …

```
Guid empty = Guid.Empty;
Console.WriteLine(empty);

Guid guid = Guid.NewGuid();
Console.WriteLine(guid);

string guidS = guid.ToString();
Console.WriteLine(guidS);

bool equals = guid.Equals(empty);
Console.WriteLine(equals);
```

```
00000000-0000-0000-0000-000000000000

957deedf-b55e-4d68-99ca-28f043d785f4

957deedf-b55e-4d68-99ca-28f043d785f4

False
```