C#

The good, the bad and the ugly

Thorsten Kansy (tkansy@dotnetconsulting.eu)

Thorsten Kansy

Freier Consultant, Software Architekt,
Entwickler, Trainer & Fachautor

DOTNETCONSULTING
by Thorsten Kansy

Agenda

- The Good
- The Bad
- The Ugly

- Warum ist es ein Problem?

- Wie sieht die Lösung aus?

I  C#

DOTNETCONSULTING
by Thorsten Kansy

The Good

Tuppel

C# 7

Rückgabe von mehr als einem Wert

```
static (int, bool) parseAsInt(string input)
{
    bool success = int.TryParse(input, out int result);

    return (result, success);
}
```

Out-Variablen

C# 7

Deklaration & out-Zusatz in einem

```
static (int, bool) parseAsInt(string input)
{
    bool success = int.TryParse(input, out int result);

    return (result, success);
}
```

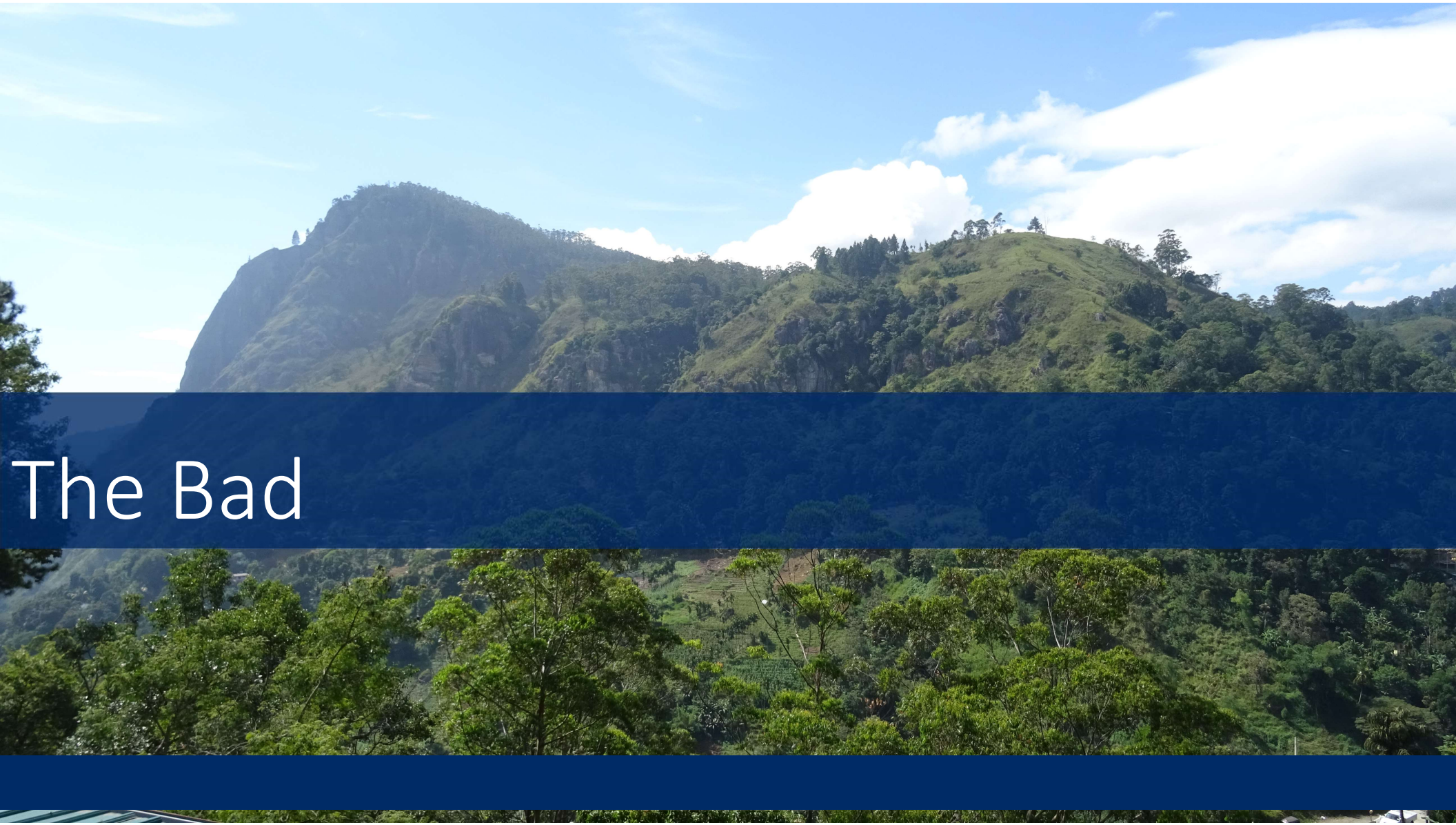
Switch mit Pattern

C# 7

```
switch (item)
{
    case 0:
        break;
    case int val:
        sum += val;
        break;
    case IEnumerable<object> subList when subList.Any():
        sum += DiceSum(subList);
        break;
    case IEnumerable<object> subList:
        break;
    case null:
        break;
}
```

C# 8

```
var area = figure switch
{
    Line _      => 0,
    Rectangle r => r.Width * r.Height,
    Circle c    => Math.PI * c.Radius * c.Radius,
    _           => throw new UnknownFigureException(figure)
};
```



The Bad

Auf Nummer sicher gehen

Gut, aber bis zu welchem Grad?

- Muß der Compiler getestet werden?
- Ein/das Framework testen?

```
static void PlayItSafe(int a, int b)
{
    int c = a + b;
    Debug.Assert(a + b == c);

    …
}
```

Kommentare

„Warum“ statt „was“ beschreiben

Zusammen mit Code pflegen

```
static void Comments(int a, int b)
{
    // Prüfen, ob a größer b ist
    if (a > b)
    {
        // "a größer b" im Debug-Fenster ausgeben
        Console.WriteLine("a größer b");
    }
}
```

Sprechende Namen

Wichtiger als Kommentare? Nein

Gut in Kombination mit Kommentaren? Sicher

```
static void DescriptiveNames(int arg1, int arg2, int arg3)
{
    int p1, p2, p3;

    …
}
```

Granularität

Gut, aber bis zu welchem Grad?

```
static int AddNumbers(int a, int b)
{
    return a + b;
}
```

```
static int AddNumbers(int a, int b)
{
    // return a + b;
    return 0;
}
```

Klarheit

Wenn man den (einfachen) Typ doch kennt

Nur bei langen Typen-Bezeichnungen

```
static void VarOrNotVar(int a, int b)
{
    var sum = a + b;
    …
}
```

No Magic Numbers

Verschleitert die Bedeutung

Besser Konstanten mit guten Namen

```
static void ProcessData(Data data)
{
    if (data.status == 451)
    {
        // ...
    }
}
```

The Ugly

Es gibt Git, SVN & Co.

Keine Kommentare nur weil Sie grün sind

```
//protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
//{
//    if (!optionsBuilder.IsConfigured)
//    {
//        // Konfiguration auf Azure Comos DB Emulator
//        string endPoint = @"https://localhost:8081";
//        string authKey = @"C2y6yDjf5/R+ob0N8A7Cgv30VRDJIWEHLM+4QDU5DE2nQ9nDuVTqobD4b8mGGyP
//        string databaseName = @"EFDatabaseOrders";
//        optionsBuilder
//            .EnableSensitiveDataLogging(true)
//            .ConfigureWarnings(c =>
//            {
//                // c.Throw(???);
//            })
//            .UseCosmos(endPoint, authKey, databaseName);
//    }

//    base.OnConfiguring(optionsBuilder);
//}

//protected override void OnModelCreating(ModelBuilder mb)
//{
//    mb.Entity<Order>(e =>
//    {
//        e.ToContainer("Orders");
//        // e.Metadata.Cosmos().ContainerName = "Orders";
```

Kommentare 2

Offentsichtliches muss nicht beschrieben werden

Zusammen mit Code pflegen!

```
/// <summary>
/// Parses input as integer.
/// </summary>
/// <param name="input">The input.</param>
/// <returns>The value as integer.</returns>
static (int, bool) parseAsInt(string input)
{
    …
}
```

False Constants

Oft nicht gepflegte Namen

Leider oft nicht so leicht erkennbar

```
static void func()
{
    const bool TRUE = false;
    const bool FALSE = true;
    …
}
```

Widersprüchliches vermeiden

Ja, was denn nun?

```
static int AddNumbers(int a, int b)
{
    return a - b;
}
```

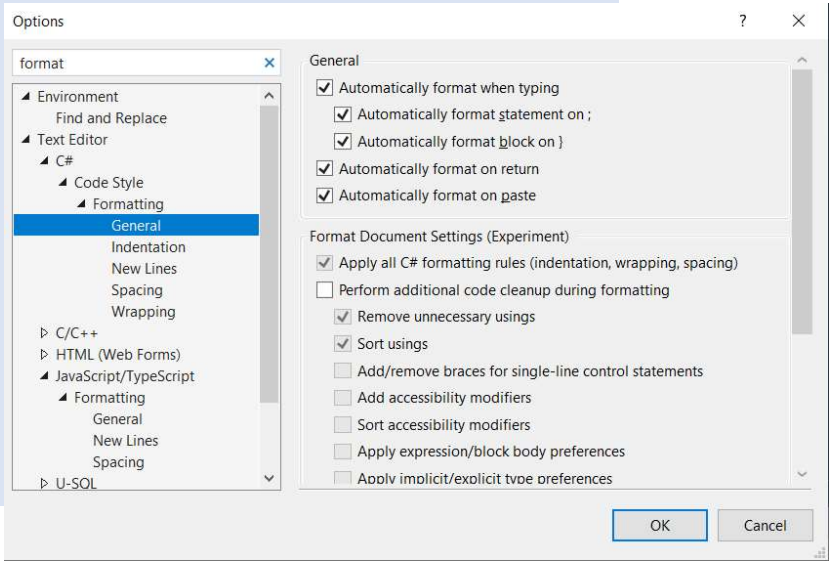
Formatierungen

Erschweren das Lesen

Nicht gegen den Editor formatieren

```
static void Comments(int a, int b)
{
if (a > b)
{
Console.WriteLine("a größer b");
}


}
```



DOTNETCONSULTING
by Thorsten Kansy

Lustig: vielleicht? Problem: ja!

Warum ist solcher Code ein Problem?

- Schlechter Code lenkt ab
- Höhere Kosten
  - Entwicklung
  - Wartung
    - Verständnis
    - Anpassung
- Höherer Stress (für Entwickler)
- Höhere Fehleranfälligkeit

Projektstrukturen

- Aufteilung in wieviele Projekte?
  - Monolith vs Nano-Projekte
- Sind 902 Projekte eventuell zu viele?
- Wiederverwendbarkeit um jeden Preis?

Was ist mit Design-Pattern?

- Sinnvoll?
  - Ja, das Rad nicht neu erfinden!
- Aber nicht als Selbstzweck!
- Pattern vs Kreativität

Fazit

- Jeder Code ohne Sinn ist überflüssig
  - Wirklich? Wirklich!
- KISS-Prinzip
  - Keep it simple, stupid
- Pfadfinder-Prinzip

Fragen?

Links

www.dotnetconsulting.eu

@Tkansy

tkansy@dotnetconsulting.eu

DOTNETCONSULTING
by Thorsten Kansy