Konsolenparameter

in C#

Übersicht

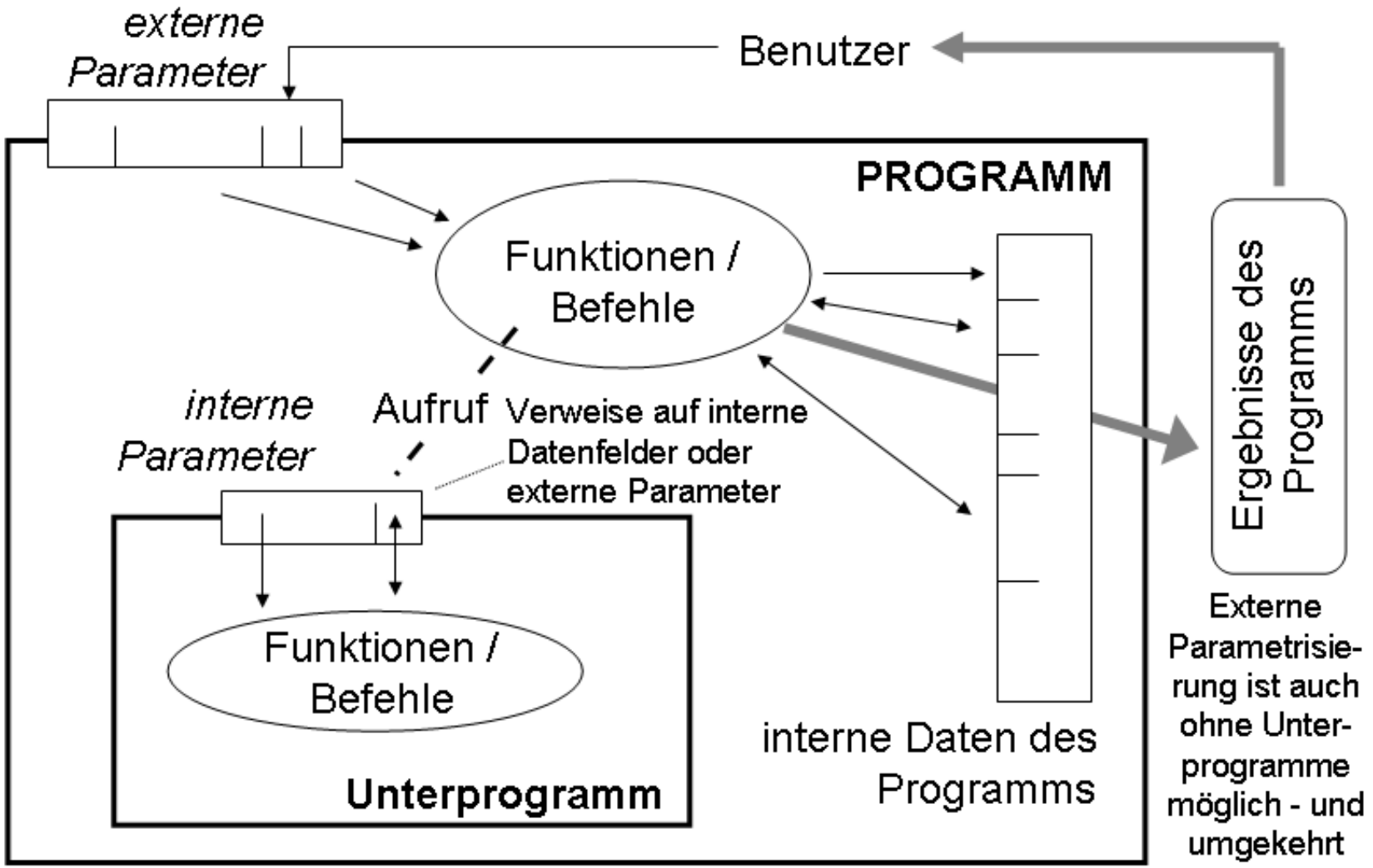
- Kommandozeilenargumente
- Prinzip der Parametrisierung
- Einlesen und Ausgeben von Kommandozeilenargumenten
- Faktorielle berechnen (iterativ)
- Taschenrechner mit Kommandozeilenparameter

Kommandozeilenargumente

- Umgang mit der Kommandozeile wichtig bei älteren Programmen
- Bei Konsolenprogramme für Linux/UNIX oder MS-DOS (Eingabeaufforderung) sind Kommandozeilenparameter immer noch ein wichtiges Konzept

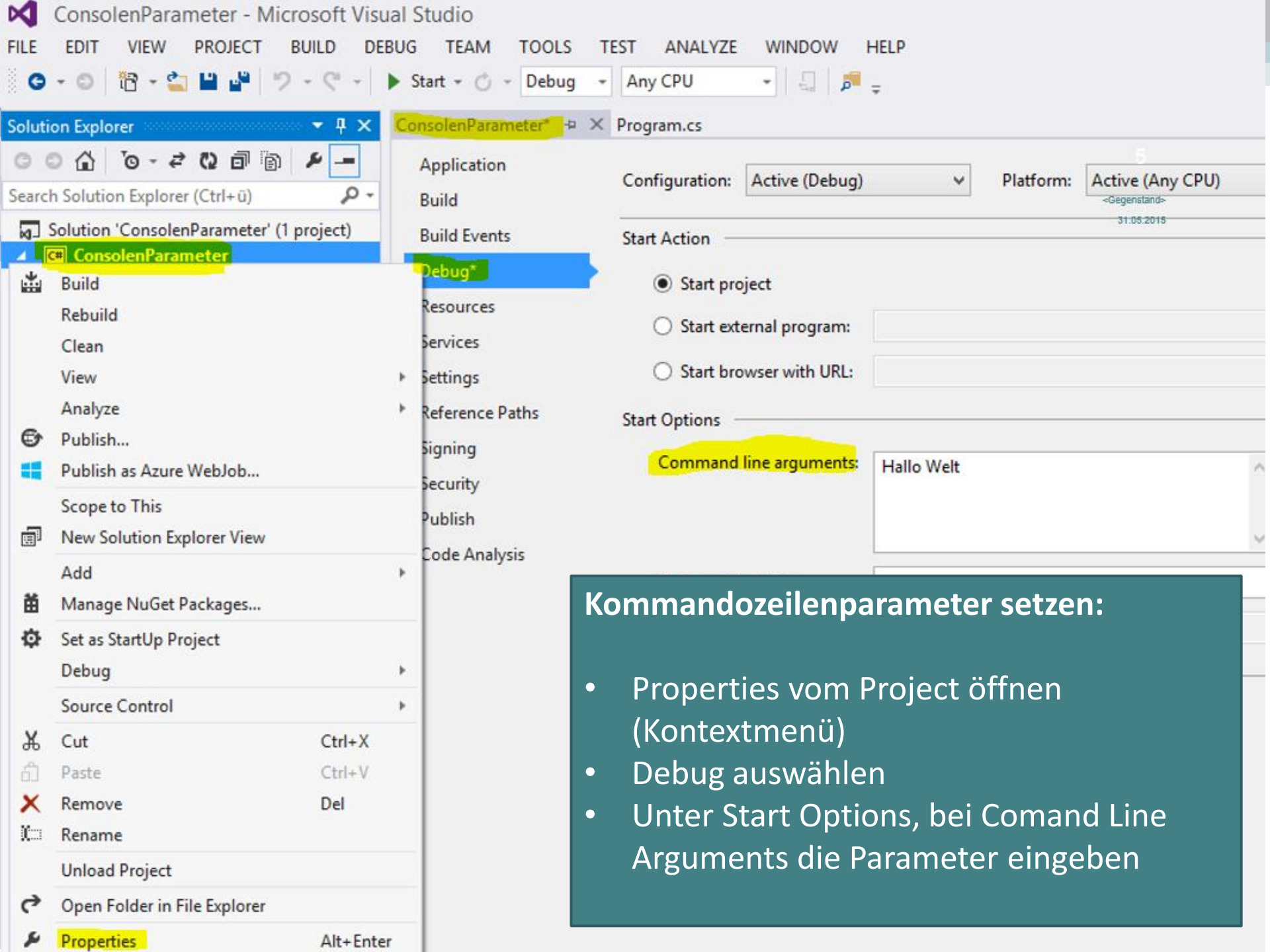
- Keine grafische Oberfläche -> Kommandozeile wichtigste Schnittstelle zwischen Anwender & Programm

Prinzip der Parametrisierung



Aufgabe: Einlesen von Parametern

- Erstelle eine neue Konsolenanwendung „ConsolenParameter“
- Öffne die Eigenschaften (Properties) vom Projekt und setze unter Debug die Eingabeparameter (Command Line Arguments)
  - Anleitung siehe Screenshot
- Erstelle ein Programm, das überprüft ob Argumente übergeben wurden -> falls nicht Exit + Fehlermeldung
- Ansonsten gib die Argumente in der Konsole aus.



Kommandozeilenparameter setzen:

- Properties vom Project öffnen (Kontextmenü)
- Debug auswählen
- Unter Start Options, bei Comand Line Arguments die Parameter eingeben

Quellcode + Ausgabe

```
namespace ConsolenParameter
{
    class Program
    {
        static int Main(string[] args)
        {
            if (args.Length == 0)
            {
                Console.WriteLine("Bitte Argumente angeben.");
                return 1;
            }

            for (int i = 0; i < args.Length; i++)
                Console.WriteLine(i + ": " + args[i]);
            return 0;
        }
    }
}
```

```
0: Hallo
1: Welt
Drücken Sie eine beliebige Taste . . .
```

Aufruf über die Eingabeaufforderung

- Navigiere im Explorer zu deinem C# Programm:
  - ConsolenParameter\bin\Debug\ConsolenParameter.exe

```
Bsp\ConsolenParameter\ConsolenParameter\bin\Debug

31.05.2015  11:05    <DIR>          .
31.05.2015  11:05    <DIR>          ..
31.05.2015  11:33             5 120 ConsolenParameter.exe
31.05.2015  11:04               187 ConsolenParameter.exe.config
31.05.2015  11:33            13 824 ConsolenParameter.pdb
31.05.2015  11:20            23 168 ConsolenParameter.vshost.exe
31.05.2015  11:04               187 ConsolenParameter.vshost.exe.config
18.06.2013  14:28               490 ConsolenParameter.vshost.exe.manifest
               6 Datei(en),         42 976 Bytes
               2 Verzeichnis(se), 44 043 821 056 Bytes frei
```

- Starte das Programm mit und ohne Parameter:

```
D:\home\bibi\Aktuell\HTL\HTL_Krems\Softwareentwicklung\C_Sharp_Bsp\ConsolenParam
eter\ConsolenParameter\bin\Debug>ConsolenParameter.exe
Bitte Argumente angeben.

D:\home\bibi\Aktuell\HTL\HTL_Krems\Softwareentwicklung\C_Sharp_Bsp\ConsolenParam
eter\ConsolenParameter\bin\Debug>ConsolenParameter.exe a b c d e f g
0: a
1: b
2: c
3: d
4: e
5: f
6: g
```

Erstelle ein Programm für Faktorielle

- Benutzt wird das Programm mit:
  - Programmname Zahl
  - (Erwünschte Eingabe >= 0 und <= 20)

```
Please enter a numeric argument.
Usage: Factorial <num>


     ConsolenParameter\bin\Debug>ConsolenParameter.exe 7
The Factorial of 7 is 5040.


     ConsolenParameter\bin\Debug>ConsolenParameter.exe 2
The Factorial of 2 is 2.


     ConsolenParameter\bin\Debug>ConsolenParameter.exe 5
The Factorial of 5 is 120.


     ConsolenParameter\bin\Debug>ConsolenParameter.exe 3
The Factorial of 3 is 6.
```

Explizite Fakultätswerte von 0 bis 20

0!	1
1!	1
2!	2
3!	6
4!	24
5!	120
6!	720
7!	5.040
8!	40.320
9!	362.880
10!	3.628.800
11!	39.916.800

Berechnung mit der Funktion

```
public static long Factorial(int n)
{
    // Test for invalid input
    if ((n < 0) || (n > 20))
    {
        return -1;
    }

    // Calculate the factorial iteratively rather than recursively:
    long tempResult = 1;
    for (int i = 1; i <= n; i++)
    {
        tempResult *= i;
    }
    return tempResult;
}
```

Main Teil 1

```
static int Main(string[] args)
{
    // Test if input arguments were supplied:
    if (args.Length == 0)
    {
        System.Console.WriteLine("Please enter a numeric argument.");
        System.Console.WriteLine("Usage: Factorial <num>");
        return 1;
    }

    // Try to convert the input arguments to numbers. This will throw
    // an exception if the argument is not a number.
    // num = int.Parse(args[0]);
    int num;
    bool test = int.TryParse(args[0], out num);
    if (test == false)
    {
        System.Console.WriteLine("Please enter a numeric argument.");
        System.Console.WriteLine("Usage: Factorial <num>");
        return 1;
    }
```

Main Teil 2

```
    // Calculate factorial.
    long result = Factorial(num);

    // Print result.
    if (result == -1)
        System.Console.WriteLine("Input must be >= 0 and <= 20.");
    else
        System.Console.WriteLine("The Factorial of {0} is {1}.", num, result);

    return 0;
}

// If 3 is entered on command line, the
// output reads: The factorial of 3 is 6.
```

Erstelle einen Taschenrechner

- Der über die Konsole
  - <Zahl><Operand><Zahl> beliebig oft einlesen kann
  - Und das Ergebnis ausgibt

  - Bsp:
  - 2 + 3 * 4 / 5

Calc mit Kommandozeilenparameter

- Es sollen mindestens 4 Parameter übergeben werden

```
public static int Main (String[] args)
{
    long number, result;

    if (args.Length < 3)
    {
        Console.WriteLine("Benötige mindestens 3 Argumente!");
        Console.WriteLine("Aufruf: calc.exe <zahl><op><zahl> ...");
        return 1;
    }
```

Berechnung für beliebig viele Zahlen

```
/* 1.Zahl in einen Integer konvertieren*/
result = Convert.ToInt32(args[0]);
for (int i = 1; i < args.Length; i += 2 )
{
    number = Convert.ToInt32(args[i + 1]);
    if (args.Length > i)
    {
        switch (args[i])
        {
            case "+":
                result += number;
                break;
            case "-":
                result -= number;
                break;
            case "*":
                result *= number;
                break;
            case "/":
                result /= number;
                break;
            default:
                Console.WriteLine("Ungültiger Operand!");
                break;
        }
    }
}
    Console.WriteLine("Ergebnis: {0}", result);
    Console.ReadKey();
return 0;
}
```