

Methoden und Parameter Modifizierer

Freitag, 13. Januar 2017

14:09

```
public int myFunction(int a, out string s, ref double  
d, params double[] d);
```

out	<p>Ausgabeparameter</p> <ul style="list-style-type: none">• müssen in der Methode zugewiesen werden• müssen nicht vor Aufruf der Methode initialisiert werden• pass-by-reference <p>Beispiel</p> <pre>public static int myFunction(out double d) { d = 1; return 0; }</pre> <p>Aufruf:</p> <pre>double d; int i = myFunction(out d); // d=1</pre>
ref	<p>Referenzparameter</p> <ul style="list-style-type: none">• Wert-Änderungen ändern den Parameter auch nach Abarbeitung der Methode• pass-by-reference <p>Beispiel</p> <pre>public static void myFunction(ref double d) { // d=0 d = 1; }</pre> <p>Aufruf:</p> <pre>double d = 0.0; int i = myFunction(ref d);</pre>

	<pre>// d=1</pre>
params	<p>Beliebig viele Parameter</p> <ul style="list-style-type: none"> • Erlaubt beliebig viele Parameter des gleichen Typs zu übergeben ohne dass explizit ein Array erzeugt werden muss • nur ein einzelner params Modifizierer pro Methode, stets an letzter Stelle der Argumentliste <p>Beispiel</p> <pre>public static double add(params double[] d) { double sum = 0; for (int i=0; i<d.Length; i++) sum += d[i]; return sum ; }</pre> <p>Aufruf:</p> <pre>double summe = add(0.1, 0.2, 0.3);</pre>
(nichts)	<p>pass-by-value</p> <ul style="list-style-type: none"> • Wert der übergebenen Variable wird in eine neue lokale Variable kopiert <p>Beispiel</p> <pre>public static void myFunction(double d) { d = 1; }</pre> <p>Aufruf:</p> <pre>double d = 0; int i = myFunction(d); // still d = 0</pre>

Wie viele Fehler finden Sie ?

(Angenommen die Signatur der Funktion ist korrekt)

```
public static int myFunction(int a, out int o, ref int
r)
{
    int result = a+o+r;

    return result;
}

static void Main(string[] args)
{
    double a;

    myFunction(a, 1, 2);
}
```