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** | **Ausgabeparameter**   * müssen in der Methode zugewiesen werden * müssen nicht vor Aufruf der Methode initialisiert werden * pass-by-reference     Beispiel  **public** static int myFunction**(out** double d**){**  d **= 1;**  **return 0;**  **}**    **Aufruf:**  double d;  int i **=** myFunction**(out** d);  // d=1 |
| **ref** | **Referenzparameter**   * Wert-Änderungen ändern den Parameter auch nach Abarbeitung der Methode * pass-by-reference     Beispiel  **public** static void myFunction**(ref** double d**){**  // d=0  d **= 1;**  **}**    **Aufruf:**  double d = 0.0;  int i **=** myFunction**(ref** d);  // d=1 |
| **params** | **Beliebig viele Parameter**   * 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     Beispiel  **public** static double add**(params** double[] d**){**  double sum **=** 0**;**  **for (**int i**=**0**;** i**<**dd**.**Length**;** i**++)** sum **+=** d**[**i**];**  **return** sum **;**  **}**    **Aufruf:**  double summe **=** add**(0.1, 0.2, 0.3**); |
| **(nichts)** | **pass-by-value**   * Wert der übergebenen Variable wird in eine neue lokale Variable kopiert     Beispiel  **public** static void myFunction**(**double d**){**  d = 1;  **}**    **Aufruf:**  double d = 0;  int i **=** myFunction**(**d**)**;  // still d = 0 |

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**);**

**}**