Team: **Valeriya Koshikova, Jerome Weber**

Parametertypen

1.Benannte Parameter

2.Ref-Parameter

3.Out-Parameter

4.Standardparameter oder optionale Parameter

5. Dynamishe Parameter

6.Wertparameter



7.Parameter

1.Benannte Parameter

Mit der Hilfe Benannte Parameter kann man den Wert des Parameters entsprechend seinem Namen und nicht seiner Reihenfolge in der Methode angeben.

Beispiel:

using System;

public class GFG {

public static void addstr (string s1, string s2, string s3){



string result = s1 + s2 + s3;

Console. WriteLine ("Final string is: " + result);

}

static public void Main (){

addstr (s1: "Geeks", s2: "for", s3: "Geeks");



}

}

2.Ref-Parameter

Mit der Hilfe Ref-Parameter kann man den Wert des Parameters durch Verweis übergeben.

Beispiel:

using System;

class GFG {

public static void Main (){

string val = "Dog";

CompareValue(ref val);



Console. WriteLine(val);

}

static void CompareValue(ref string val1){



if (val1 == "Dog"){

Console.WriteLine("Matched!");

}

val1 = "Cat";

}

}

3.Out-Parameter

Mit der Hilfe Out-Parameter kann man den Wert des Parameters mehrere Werte zurückgeben.

Beispiel:

using System;

class GFG {

static public void Main(){

int num;

AddNum(out num);



Console.WriteLine("The sum of"+ " the value is: {0}",num);

}

public static void AddNum(out int num){



num = 40;

num += num;

}

}

4.Standardparameter oder optionale Parameter

Mit der Hilfe Standardparameter oder optionale Parameter kann man Argumente für einige Parameter auszuschließen.

Beispiel:

using System;



class GFG {

static public void detail (string ename,int eid,string bgrp = "A+",string dept = "Review-Team")



{

Console.WriteLine("Employee name: {0}", ename);



Console.WriteLine("Employee ID: {0}", eid);



Console.WriteLine("Blood Group: {0}", bgrp);



Console.WriteLine("Department: {0}", dept);



}

static public void Main(){

detail("XYZ", 123);

detail("ABC", 456, "B-");

detail("DEF", 789, "B+",

"Software Developer");

}

}

5. Dynamishe Parameter

Mit der Hilfe Dynamishe Parameter kann man ein neuer Parametertyp einführen, der als dynamischer Parameter bezeichnen.



Beispiel:

using System;

class GFG {

public static void mulval (dynamic val){



val \*= val;

Console.WriteLine(val);

}

static public void Main(){

mulval(30);

}

}

6.Wertparameter



Mit der Hilfe Wertparameter kann man die Übergabe von Werttypen nach Wert sagen.

Beispiel:

using System;

public class GFG {

static public void Main(){

string str1 = "Geeks";



string str2 = "geeks";



string res = addstr(str1, str2);

Console.WriteLine(res);

}

public static string addstr(string s1, string s2){

return s1 + s2;

}

}

7.Parameter



Mit der Hilfe Parameter kann man eine beliebige Anzahl von Argumenten übergeben.

Beispiel:

using System;

namespace Examples {

class Geeks {

public static int mulval(params int[] num){



int res = 1;

foreach(int j in num){

res \*= j;

}

return res;

}

static void Main(string[] args){

int x = mulval(20, 49, 56, 69, 78);

Console.WriteLine(x);

}

}

}