

# **Microsoft Technologien**

-

# **Ergänzungsbeispiele**



## **Inhaltsverzeichnis: Microsoft Technologien - Beispiele**

1. Boxing 1 .....	4
2. Boxing 2: .....	5
3. Konstruktoren 1 .....	6
4. Konstruktoren 2 .....	7
5. Parameterübergabe .....	8

# 1. Boxing 1

```
using System;
using System.Reflection;

namespace Übungsaufgaben
{
    struct Punkt
    {
        public int x;
        public int y;
    }

    class Boxing1
    {
        static void Main(string[] args)
        {
            Punkt p;
            p.x = 1;
            p.y = 2;
            Console.WriteLine("Punkt:    x={0} y={1}", p.x, p.y);
            object o = p;
            Console.WriteLine("Object:   x={0} y={1}", ((Punkt) o).x, ((Punkt)
o).y);
            Type t = p.GetType();
            FieldInfo fx = t.GetField("x");
            FieldInfo fy = t.GetField("y");
            fx.SetValue(o, 9);
            fy.SetValue(p, 8);
            Console.WriteLine("Punkt:    x={0} y={1}", p.x, p.y);
            Console.WriteLine("Object:   x={0} y={1}", ((Punkt) o).x, ((Punkt)
o).y);
            Console.ReadLine();
        }
    }
}
```

## Konsolenausgabe:

```
Punkt:   x=1 y=2
Object : x=1 y=2
Punkt :  x=1 y=2
Object : x=9 y=2
```

## 2. Boxing 2:

```
using System;

namespace Übungsaufgaben
{
    class Boxing2
    {
        public static void Set(object x, ref int y, int z)
        {
            x = ((int) x) + 1;
            y++;
            z++;
            Console.WriteLine("x={0} y={1} z={2}", x, y, z);
        }

        static void Main(string[] args)
        {
            int x=0, y=0, z=0;
            Set(x, ref y, z);
            Console.WriteLine("x={0} y={1} z={2}", x, y, z);
            Console.ReadLine();
        }
    }
}
```

### **Konsolenausgabe:**

```
x=1 y=1 z=1
x=0 y=1 z=0
```

### 3. Konstruktoren 1

```
using System;

namespace Übungsaufgaben
{
    public interface IWord
    {
        void Print();
    }

    public abstract class Base : IWord
    {
        public Base() { Console.Write(Get()); }
        static Base() { Console.Write("A "); }
        public virtual void Print() { Console.Write("B "); }
        protected virtual string Get() { return("C "); }
        protected static string msg = "D ";
    }

    public class Derived : Base
    {
        static Derived() { Console.Write(Derived.msg); }
        public Derived() { ((IWord) this).Print(); }
        public new virtual void Print() { Console.Write("E "); }
        protected override string Get() { return("F "); }
    }

    public class More : Derived, IWord
    {
        public override void Print() { Console.Write("G "); }
        void IWord.Print() { Console.Write("H "); }
        protected override string Get() { return("I "); }
    }

    class Konstruktoren1
    {
        static void Main(string[] args)
        {
            Console.Write("1: ");
            Derived d = new Derived();

            Console.Write("\n2: ");
            ((Base) d).Print();

            Console.Write("\n3: ");
            More m = new More();

            Console.Write("\n4: ");
            ((Derived) m).Print();

            Console.ReadLine();
        }
    }
}
```

### Konsolenausgabe:

1: A D F B  
2: B  
3: I H  
4: G

## 4. Konstruktoren 2

```
using System;

namespace Übungsaufgaben
{
    public class Klasse
    {
        public static int a = Klasse.b;
        public static int b = 1;
        public static int c = Klasse.b;

        static Klasse()
        {
            Klasse.b = 2;
        }
    }

    class Konstruktoren2
    {
        static void Main(string[] args)
        {
            Console.WriteLine(Klasse.a);
            Console.WriteLine(Klasse.b);
            Console.WriteLine(Klasse.c);
            Console.ReadLine();
        }
    }
}
```

### Konsolenausgabe:

0  
2  
1

## 5. Parameterübergabe

```
using System;

namespace Übungsaufgaben
{
    class Parameterübergabe1
    {
        private static void DoIt(int i, ref int j)
        {
            j += i;
            i--;
            Console.WriteLine("A1: i={0} j={1}", i, j);
            if (i>0) DoIt(i, ref j);
            Console.WriteLine("A2: i={0} j={1}", i, j);
        }

        static void Main(string[] args)
        {
            int i = 3;
            int j = 0;
            Console.WriteLine("Vor DoIt(): i={0} j={1}", i, j);
            DoIt(i, ref j);
            Console.WriteLine("Nach DoIt(): i={0} j={1}", i, j);
            Console.ReadLine();
        }
    }
}
```

### Konsolenausgabe:

```
Vor DoIt(): i=3 j=0
A1: i=2 j=3
A1: i=1 j=5
A1: i=0 j=6
A2: i=0 j=6
A2: i=1 j=6
A2: i=2 j=6
Nach DoIt(): i=3 j=6
```