

1 Die Math-Klasse

Mit Hilfe der Klasse `System.Math` können wir mathematische Funktionen in unserem Programm verwenden. Dazu stellt uns diese Klasse verschiedenste statische Methoden zur Verfügung.

Fangen wir am besten einfach direkt damit an, einige interessante Methoden zu betrachten.

1.1 Die Methode `Math.Abs()`

Diese Methode gibt uns den Absolutwert (Betrag) einer beliebigen Zahl zurück.

Beispiel:

```
using System;

namespace Mathe
{
    class Program
    {
        static void Main(string[] args)
        {
            int zahl1 = 20;
            int zahl2 = -20;

            Console.WriteLine(Math.Abs(zahl1));
            Console.WriteLine(Math.Abs(zahl2));

            Console.ReadKey();
        }
    }
}
```

Ausgabe:

```
20
20
```

Der übergebene Parameter kann hierbei von einem beliebigen Datentyp für Zahlen sein.

1.2 Die Methode `Math.Acos()`

Diese Methode berechnet den Arkuskosinus einer angegebenen Zahl vom Datentyp `double`. Das heißt, dass der Kosinus des Ergebnisses der übergebenen Zahl entspricht.

Der Rückgabewert gibt einen Winkel im Bogenmaß vom Datentyp *double* an.
Ein Beispiel dazu folgt nun.

```
using System;

namespace Mathe
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(Math.Acos(1));
            Console.WriteLine(Math.Acos(0));

            Console.ReadKey();
        }
    }
}
```

Ausgabe:

```
0
1,5707963267949
```

Allgemein gilt: Möchte man einen Winkel vom Bogenmaß in das Gradmaß umwandeln, so multipliziert man diesen einfach mit dem Faktor $\frac{180^\circ}{\pi}$.

1.3 Die Methode Math.Asin()

Diese Methode berechnet den Arkussinus einer angegebenen Zahl vom Datentyp *double*. Das heißt, dass der Sinus des Ergebnisses der übergebenen Zahl entspricht.

Der Rückgabewert gibt einen Winkel im Bogenmaß vom Datentyp *double* an.

Beispiel:

```
using System;

namespace Mathe
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(Math.Asin(1));
            Console.WriteLine(Math.Asin(0));

            Console.ReadKey();
        }
    }
}
```

```
}  
}
```

Ausgabe:

```
1,5707963267949  
0
```

1.4 Die Methode Math.Atan()

Diese Methode berechnet den Arkustangens einer angegebenen Zahl vom Datentyp `double`. Das heißt, dass der Tangens des Ergebnisses der übergebenen Zahl entspricht.

Der Rückgabewert gibt einen Winkel im Bogenmaß vom Datentyp `double` an.

Beispiel:

```
using System;  
  
namespace Mathe  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            Console.WriteLine(Math.Atan(1));  
            Console.WriteLine(Math.Atan(0));  
  
            Console.ReadKey();  
        }  
    }  
}
```

Ausgabe:

```
0,785398163397448  
0
```

1.5 Die Methode Math.Ceiling()

Diese Methode entspricht der oberen Gaußklammer. Das bedeutet, die übergebene Zahl vom Datentyp `double` oder `decimal` wird immer aufgerundet zurückgegeben.

Beispiel:

```
using System;

namespace Mathe
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(Math.Ceiling(1.6));
            Console.WriteLine(Math.Ceiling(2.2));

            Console.ReadKey();
        }
    }
}
```

Ausgabe:

```
2
3
```

1.6 Die Methode Math.Cos()

Diese Methode liefert den Kosinus einer angegebenen Zahl vom Datentyp *double* zurück.

Beispiel:

```
using System;

namespace Mathe
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(Math.Cos(0));
            Console.WriteLine(Math.Cos(1));

            Console.ReadKey();
        }
    }
}
```

Ausgabe:

```
1
0,54030230586814
```

1.7 Die Methode Math.Cosh()

Diese Methode liefert den Kosinus hyperbolicus einer angegebenen Zahl vom Datentyp *double* zurück.

Beispiel:

```
using System;

namespace Mathe
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(Math.Cosh(0));
            Console.WriteLine(Math.Cosh(1));

            Console.ReadKey();
        }
    }
}
```

Ausgabe:

```
1
1,54308063481524
```

1.8 Die Methode Math.Exp()

Liefert den Wert der *e*-Funktion hoch einem angegebenen Exponenten vom Datentyp *double* zurück. Beispiel:

```
using System;

namespace Mathe
{
    class Program
    {
        static void Main(string[] args)
        {
            //  $e^0 = 1$ 
            Console.WriteLine(Math.Exp(0));
            //  $e^1 = e = 2,71\dots$ 
        }
    }
}
```

```

        Console.WriteLine(Math.Exp(1));

        Console.ReadKey();
    }
}

```

Ausgabe:

```

1
2,71828182845905

```

1.9 Die Methode Math.Floor()

Diese Methode entspricht der unteren Gaußklammer. Das bedeutet, die übergebene Zahl vom Datentyp *double* oder *decimal* wird immer abgerundet zurückgegeben.

Beispiel:

```

using System;

namespace Mathe
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(Math.Floor(1.6));
            Console.WriteLine(Math.Floor(2.2));

            Console.ReadKey();
        }
    }
}

```

Ausgabe:

```

1
2

```

1.10 Die Methode Math.Log()

Diese Methode gibt den natürlichen Logarithmus (zur Basis *e*) der übergebenen Zahl vom Datentyp *double* zurück, wenn man ihr lediglich einen Parameter übergibt.

Beispiel:

```
using System;

namespace Mathe
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(Math.Log(1));
            Console.WriteLine(Math.Log(10));

            Console.ReadKey();
        }
    }
}
```

Ausgabe:

```
0
2,30258509299405
```

Übergibt man dieser Methode hingegen einen zweiten Parameter vom Datentyp *double*, so gibt dieser die Basis an.

Beispiel:

```
using System;

namespace Mathe
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(Math.Log(1, 2));
            Console.WriteLine(Math.Log(10, 2));

            Console.ReadKey();
        }
    }
}
```

Ausgabe:

```
0
3,32192809488736
```

1.11 Die Methode Math.Log10()

Diese Methode liefert uns den Logarithmus einer übergebenen Zahl vom Datentyp `double` zur Basis 10 zurück.

Beispiel:

```
using System;

namespace Mathe
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(Math.Log10(1));
            Console.WriteLine(Math.Log10(10));

            Console.ReadKey();
        }
    }
}
```

Ausgabe:

```
0
1
```

1.12 Die Methode Math.Max()

Diese Methode gibt uns die größere von zwei übergebenen Zahlen zurück. Die Zahlen können von einem beliebigen Datentyp für Zahlen sein, jedoch müssen beide Zahlen denselben Datentyp haben. Beispiel:

```
using System;

namespace Mathe
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(Math.Max(0, 1));
            Console.WriteLine(Math.Max(1, 10));

            Console.ReadKey();
        }
    }
}
```


Ausgabe:

```
1
10
```

1.13 Die Methode Math.Min()

Diese Methode gibt uns die kleinere von zwei übergebenen Zahlen zurück. Die Zahlen können von einem beliebigen Datentyp für Zahlen sein, jedoch müssen beide Zahlen denselben Datentyp haben.

Beispiel:

```
using System;

namespace Mathe
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(Math.Min(0, 1));
            Console.WriteLine(Math.Min(1, 10));

            Console.ReadKey();
        }
    }
}
```

Ausgabe:

```
0
1
```

1.14 Die Methode Math.Pow()

Diese Methode potenziert den ersten Parameter zu einem angegebenen Exponenten (zweiter Parameter). Beide Parameter sind vom Datentyp *double*. Beispiel:

```
using System;

namespace Mathe
{
    class Program
    {
        static void Main(string[] args)
        {
```

```

    Console.WriteLine(Math.Pow(10, 2));
    Console.WriteLine(Math.Pow(10, 3));

    Console.ReadKey();
}
}
}

```

Ausgabe:

```

100
1000

```

1.15 Die Methode Math.Round()

Diese Methode rundet eine Kommazahl vom Datentyp *double* oder *decimal*, wenn man ihr nur einen Parameter übergibt.

Beispiel:

```

using System;

namespace Mathe
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(Math.Round(10.2));
            Console.WriteLine(Math.Round(10.5));

            Console.ReadKey();
        }
    }
}

```

Ausgabe:

```

10
10

```

Wie man sieht, wurde die Zahl 10.5 nicht auf 11, sondern auf 10 gerundet. Für den Fall, dass der übergebene Wert genau in der Mitte von zwei Zahlen liegt, kann man einen weiteren Parameter angeben, um zu bestimmen, wie gerundet werden soll:

```

using System;

```

```
namespace Mathe
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(Math.Round(10.5, MidpointRounding.AwayFromZero));
            Console.WriteLine(Math.Round(10.5, MidpointRounding.ToEven));
            Console.WriteLine(Math.Round(10.5));

            Console.ReadKey();
        }
    }
}
```

Ausgabe:

```
11
10
10
```

Dabei gibt es nur die beiden eben gezeigten Möglichkeiten:

- `MidpointRounding.AwayFromZero`: Rundet auf die nächste größere Ganzzahl.
- `MidpointRounding.ToEven`: Rundet auf die nächste gerade Ganzzahl.

Darüber hinaus können wir hinter unserem angegebenen Wert noch einen weiteren Parameter vom Datentyp `int` angeben, der angibt, wie genau das Ergebnis gerundet werden soll (Anzahl der Nachkommastellen).

Beispiel:

```
using System;

namespace Mathe
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(Math.Round(10.555555555, 5,
                MidpointRounding.AwayFromZero));

            Console.WriteLine(Math.Round(10.555555555, 5,
                MidpointRounding.ToEven));

            Console.WriteLine(Math.Round(10.555555555, 5));

            Console.ReadKey();
        }
    }
}
```

```
}
```

Ausgabe:

```
10,55556
10,55556
10,55556
```

1.16 Die Methode Math.Sign()

Diese Methode gibt eine Zahl zurück, die das Vorzeichen einer übergebenen Zahl von einem beliebigen Datentyp für Zahlen angibt.

Dabei gibt es drei mögliche Rückgabewerte:

- ▶ -1: Die Zahl ist negativ (kleiner als Null).
- ▶ 0: Die Zahl ist gleich Null.
- ▶ 1: Die Zahl ist positiv (größer als Null).

Beispiel:

```
using System;

namespace Mathe
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(Math.Sign(-5));
            Console.WriteLine(Math.Sign(0));
            Console.WriteLine(Math.Sign(5));

            Console.ReadKey();
        }
    }
}
```

```
-1
0
1
```

1.17 Die Methode Math.Sin()

Diese Methode liefert den Sinus einer angegebenen Zahl vom Datentyp *double* zurück. Beispiel:

```
using System;

namespace Mathe
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(Math.Sin(0));
            Console.WriteLine(Math.Sin(1));

            Console.ReadKey();
        }
    }
}
```

Ausgabe:

```
0
0,841470984807897
```

1.18 Die Methode Math.Sinh()

Diese Methode liefert den Sinus hyperbolicus einer angegebenen Zahl vom Datentyp *double* zurück.

Beispiel:

```
using System;

namespace Mathe
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(Math.Sinh(0));
            Console.WriteLine(Math.Sinh(1));

            Console.ReadKey();
        }
    }
}
```

Ausgabe:

```
0
1,1752011936438
```

1.19 Die Methode Math.Sqrt()

Diese Methode berechnet die Quadratwurzel einer angegebenen Zahl vom Datentyp *double*.

Beispiel:

```
using System;

namespace Mathe
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(Math.Sqrt(4));
            Console.WriteLine(Math.Sqrt(9));

            Console.ReadKey();
        }
    }
}
```

Ausgabe:

```
2
3
```

1.20 Die Methode Math.Tan()

Diese Methode liefert den Tangens einer angegebenen Zahl vom Datentyp *double* zurück.

Beispiel:

```
using System;

namespace Mathe
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(Math.Tan(0));
        }
    }
}
```

```
        Console.WriteLine(Math.Tan(1));  
        Console.ReadKey();  
    }  
}
```

Ausgabe:

```
0  
1,5574077246549
```

1.21 Die Methode Math.Tanh()

Diese Methode liefert den Tangens hyperbolicus einer angegebenen Zahl vom Datentyp *double* zurück.

Beispiel:

```
using System;  
  
namespace Mathe  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            Console.WriteLine(Math.Tanh(0));  
            Console.WriteLine(Math.Tanh(1));  
  
            Console.ReadKey();  
        }  
    }  
}
```

Ausgabe:

```
0  
0,761594155955765
```

1.22 Die Methode Math.Truncate()

Diese Methode schneidet den Nachkommaanteil einer Kommazahl vom Datentyp *double* oder *decimal* ab.

Beispiel:

```

using System;

namespace Mathe
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(Math.Truncate(1.234));
            Console.WriteLine(Math.Truncate(1.856));

            Console.ReadKey();
        }
    }
}

```

Ausgabe:

```

1
1

```

1.23 Felder der Math-Klasse

Die Klasse `Math` beinhaltet außerdem noch zwei Felder zur Darstellung der Konstanten e und π .

Diese können folgendermaßen verwendet werden:

```

using System;

namespace Mathe
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(Math.E);
            Console.WriteLine(Math.PI);

            Console.ReadKey();
        }
    }
}

```

Ausgabe:

2,71828182845905

3,14159265358979

Verweise

- »Math-Klasse (System)«, [https://msdn.microsoft.com/de-de/library/system.math\(v=vs.110\).aspx](https://msdn.microsoft.com/de-de/library/system.math(v=vs.110).aspx),
03.10.2017