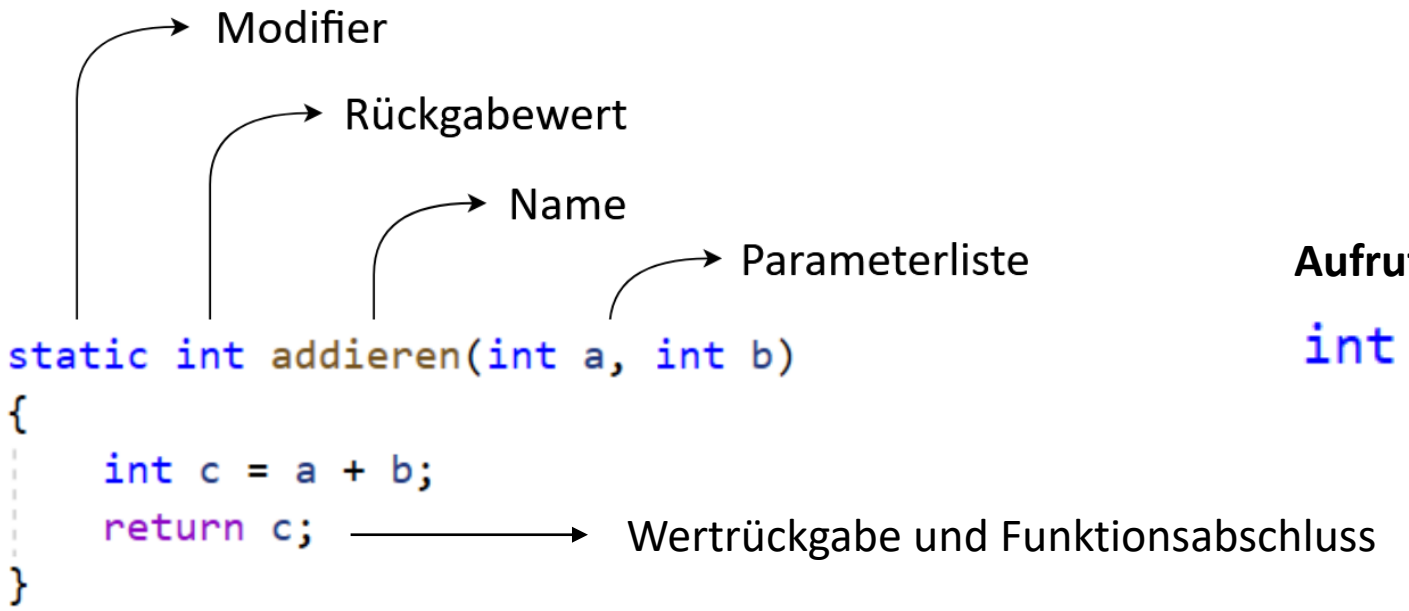
# Funktionen

# Methoden (Funktionen)

```
static void Main(string[] args)
{

    int zahl1 = 1;
    int zahl2 = 2;
    int summe = zahl1 + zahl2;
    Console.WriteLine($"Die Summe von {zahl1} und {zahl2} ist {summe}.");

    zahl1 = 5;
    zahl2 = 9;
    summe = zahl1 + zahl2;
    Console.WriteLine($"Die Summe von {zahl1} und {zahl2} ist {summe}.");

    zahl1 = 2;
    zahl2 = 7;
    summe = zahl1 + zahl2;
    Console.WriteLine($"Die Summe von {zahl1} und {zahl2} ist {summe}.");

    Console.ReadKey();
}
```

```
static void Main(string[] args)
{
    SummeAusgeben(1, 2);
    SummeAusgeben(5, 9);
    SummeAusgeben(2, 7);
    Console.ReadKey();
}

3 Verweise
static void SummeAusgeben(int zahl1, int zahl2)
{
    int summe = zahl1 + zahl2;
    Console.WriteLine($"Die Summe von {zahl1} und {zahl2} ist {summe}.");
}
```

# Methoden (Funktionen)




**Aufruf:**

```
int summe = addieren(1, 2);
```

# Methoden (Funktionen)

- Modifikatoren (Modifiers) definieren die Zugriffsmöglichkeiten auf oder den Zustand der Funktion

  z.B.: `static` `public` `private`

- Der Rückgabewert gibt den Datentyp der Rückgabe an

  z.B.: `int` `Random` `void`

- Über den Bezeichner wird die Funktion aufgerufen

- Die Parameterliste gibt die benötigten Übergabeparameter mit Datentyp und Bezeichner an

# Überladung von Methoden

- ein Funktionsname kann in derselben Klasse mehrfach benutzt werden
- jede Funktion muss eine einmalige Parameterliste haben
- die Rückgabewerte können unterschiedlich sein

```
int addieren(int a, int b)
{
    int summe = a + b;
    return summe;
}
```

```
int addieren(int a, int b, int c)
{
    int summe = a + b + c;
    return summe;
}
```

# Spezielle Parametertypen

- Automatische Arrays: Aus beliebig vielen Übergaben eines Typs wird automatisch ein Array erstellt

```
public static int BildeSumme(params int[] summanden) {…}
```

Aufrufbeispiele: `BildeSumme(2,5,7);` `BildeSumme(1,2,3,4,5,6,7)`

- Optionale Parameter: Ein Parameter erhält eine Vorbelegung, welche überschrieben werden kann

```
public static int Addiere(int a, int b, int c = 0) {…}
```

Aufrufbeispiele: `Addiere(2,5);` `Addiere(1,2,3)`

# Spezielle Parametertypen - out

- eine Funktion kann immer nur einen Rückgabewert haben
- mittels out kann mehr als ein Wert zurückgegeben werden
- Parameter wird in der Signatur deklariert und muss in der Funktion initialisiert werden
- der Aufruf erfolgt ebenfalls mit dem Schlüsselwort

```
int berechne(int zahl1, int zahl2, out int min, out int max, out int avg)
{
    int summe = zahl1 + zahl2;

    if (zahl1 < zahl2)
    {
        min = zahl1;
        max = zahl2;
    }
    else
    {
        min = zahl2;
        max = zahl1;
    }
    avg = summe / 2;

    return summe;
}
```

```
int min;
int max;
int avg;
int summe = berechne(3, 5, out min, out max, out avg);

//summe = 8
//min = 3
//max = 5
//avg = 4
```