

# Methoden/Funktionen

## Oder: Ordnung ins Programm

Michael Merk

Fachschaft Informatik  
HS Karlsruhe  
[kontakt@fachschaft-hska.de](mailto:kontakt@fachschaft-hska.de)

9. Februar 2012



# Übersicht

## Einführung

Problemstellung  
Lösung

## Methoden

Parameter  
Beispiele  
Methoden sind Funktionen  
Wertebereiche  
Umsetzung  
void

## Klassenvariablen

Probleme  
Beispiel  
Deklaration

## Objektorientierung

## Literatur

## Konventionen

### 1 Einführung

- Problemstellung
- Lösung

### 2 Methoden

- Parameter
- Beispiele
- Methoden sind Funktionen
- Wertebereiche
- Umsetzung
- void

### 3 Klassenvariablen

- Probleme
- Beispiel
- Deklaration

### 4 Objektorientierung

### 5 Literatur

### 6 Konventionen

# Problemstellung

Was ist an folgendem Code problematisch?

```
public class Main {  
    public static void Main() {  
        for(int i = 1; i <= 10; i++) {  
            Console.WriteLine("Hello World");  
        }  
        //weiterer Code  
        for(int i = 1; i <= 10; i++) {  
            Console.WriteLine("Hello World");  
        }  
    }  
}
```

Einführung

Problemstellung

Lösung

Methoden

Parameter

Beispiele

Methoden sind Funktionen

Wertebereiche

Umsetzung

void

Klassenvariablen

Probleme

Beispiel

Deklaration

Objektorientierung

Literatur

Konventionen

# Problemstellung

## Einführung

### Problemstellung

#### Lösung

## Methoden

### Parameter

### Beispiele

### Methoden sind Funktionen

### Wertebereiche

### Umsetzung

### void

## Klassenvariablen

### Probleme

### Beispiel

### Deklaration

## Objektorientierung

## Literatur

## Konventionen

```
public class Main {  
    public static void Main() {  
        for(int i = 1; i <= 10; i++) {  
            Console.WriteLine("Hello World");  
        }  
        //weiterer Code  
        for(int i = 1; i <= 10; i++) {  
            Console.WriteLine("Hello World");  
        }  
    }  
}
```

Doppelter Code ist immer problematisch.

- Viel Code
- Änderungen sind aufwendig
- Code ist sonst nirgendwo verwendbar
- Unübersichtlich

# Lösung

## Einführung

Problemstellung

Lösung

## Methoden

Parameter

Beispiele

Methoden sind Funktionen

Wertebereiche

Umsetzung

void

## Klassenvariablen

Probleme

Beispiel

Deklaration

## Objektorientierung

## Literatur

## Konventionen

```
public class Main {  
    public static void Main() {  
        for(int i = 1; i <= 10; i++) {  
            Console.WriteLine("Hello World");  
        }  
        //weiterer Code  
        for(int i = 1; i <= 10; i++) {  
            Console.WriteLine("Hello World");  
        }  
    }  
}
```

Doppelten Code an eine Stelle auslagern, wo er beliebig verwendet werden kann!  $\Rightarrow$  Methoden

## Methode

Ein Unterprogramm, das beliebig oft ausgeführt werden kann.

# Lösung

## Einführung

Problemstellung

Lösung

## Methoden

Parameter

Beispiele

Methoden sind Funktionen

Wertebereiche

Umsetzung

void

## Klassenvariablen

Probleme

Beispiel

Deklaration

## Objektorientierung

## Literatur

## Konventionen

```
public class Main {  
    public static void GibHelloWorldAus(){  
        for(int i = 1; i <= 10; i++) {  
            Console.WriteLine("Hello World");  
        }  
    }  
  
    public static void Main() {  
        GibHelloWorldAus();  
        //weiterer Code  
        GibHelloWorldAus();  
    }  
}
```

# Parameter

Der Code ist immernoch nicht toll. Ich möchte selber entscheiden, wie oft ich "Hello World" ausgabe.

Lösung: Anzahl der Methode mitgeben  $\Rightarrow$  Parameter

## Bei der Methode

Der Parameter bei der Methode steht in der Klammer und besteht aus dem **Datentyp** und einem **Namen**

$\Rightarrow$  `public static void gibHelloWorldAus(int anzahl) {...}`

## Beim Methodenaufruf

Der Parameter steht ebenfalls in der Klammer und besteht aus einem konkreten Wert (des festgelegten Typs!)

$\Rightarrow$  `GibHelloWorldAus(5);`

Einführung

Problemstellung

Lösung

Methoden

Parameter

Beispiele

Methoden sind Funktionen

Wertebereiche

Umsetzung

void

Klassenvariablen

Probleme

Beispiel

Deklaration

Objektorientierung

Literatur

Konventionen

# Beispiel 1

## Einführung

Problemstellung

Lösung

## Methoden

Parameter

Beispiele

Methoden sind Funktionen

Wertebereiche

Umsetzung

void

## Klassenvariablen

Probleme

Beispiel

Deklaration

## Objektorientierung

## Literatur

## Konventionen

```
public class Main {  
    public static void GibHelloWorldAus(int anzahl)  
    {  
        for(int i = 1; i <= anzahl; i++) {  
            Console.WriteLine("Hello World");  
        }  
    }  
  
    public static void Main() {  
        GibHelloWorldAus(12);  
        //weiterer Code  
        GibHelloWorldAus(8);  
    }  
}
```



# Beispiel 2

Es kann beliebig viele Parameter geben. Diese werden mit einem Komma getrennt.

```
public class Main {  
    public static void GibStringAus(int anzahl,  
        string ausgabeString){  
  
        for(int i = 1; i <= anzahl; i++) {  
            Console.WriteLine(ausgabeString);  
        }  
    }  
  
    public static void Main() {  
        GibStringAus(5, "Hello World");  
        //weiterer Code  
        GibStringAus(6, "Goodbye World");  
    }  
}
```

Einführung

Problemstellung

Lösung

Methoden

Parameter

Beispiele

Methoden sind Funktionen

Wertebereiche

Umsetzung

void

Klassenvariablen

Probleme

Beispiel

Deklaration

Objektorientierung

Literatur

Konventionen

# Methoden sind Funktionen

Hat das nicht was mit Mathe zu tun?

Ja, Methoden in Java/C# arbeiten genauso, wie Funktionen in der Mathematik.

## Funktion

Eine Funktion nimmt Werte entgegen, „macht etwas damit“ und gibt dies dann zurück.

## Beispiel

$$f(x) = x^2$$

Nimmt eine Zahl, quadriert sie und gibt das Ergebnis zurück.

Einführung

Problemstellung

Lösung

Methoden

Parameter

Beispiele

Methoden sind Funktionen

Wertebereiche

Umsetzung

void

Klassenvariablen

Probleme

Beispiel

Deklaration

Objektorientierung

Literatur

Konventionen

# Wertebereiche

## Einführung

Problemstellung  
Lösung

## Methoden

Parameter  
Beispiele  
Methoden sind Funktionen

## Wertebereiche

Umsetzung  
void

## Klassenvariablen

Probleme  
Beispiel  
Deklaration

## Objektorientierung

## Literatur

## Konventionen

### Wichtig

Welchen Datentyp haben die Werte, die in eine Funktion reingehen?

Welchen Datentyp haben die Werte, die aus der Funktion rauskommen?

⇒ Muss man vorher wissen!

### Beispiel

$$f(x) = x^2$$

**double**-Werte gehen in die Funktion rein und **double**-Werte kommen raus.

# Umsetzung

## Aufbau

```
public static Rückgabetyyp Name(Parameterotyp
    Parameternamen){
    //hier wird was gemacht
    return ergebnis; //WICHTIG
}
```

```
public class Main {
    public static double Quadriere(double x){
        return x*x;
    }

    public static void Main(){
        double ergebnis;

        for(int i=0;i<=20;i++){
            ergebnis = Quadriere((double) i);
            Console.WriteLine(ergebnis);
        }
    }
}
```

Einführung

Problemstellung

Lösung

Methoden

Parameter

Beispiele

Methoden sind Funktionen

Wertebereiche

Umsetzung

void

Klassenvariablen

Probleme

Beispiel

Deklaration

Objektorientierung

Literatur

Konventionen

# void

## Einführung

- Problemstellung
- Lösung

## Methoden

- Parameter
- Beispiele
- Methoden sind Funktionen
- Wertebereiche
- Umsetzung
- void**

## Klassenvariablen

- Probleme
- Beispiel
- Deklaration

## Objektorientierung

## Literatur

## Konventionen

### void

**void** bedeutet, dass eine Methode „nichts“ zurück gibt.  
⇒ es wird dann auch kein **return** benötigt.

### Beispiel: Main-Methode

```
public static void Main(){...}
```

Die Main-Methode ist eine normale Methode, die aber auch als Einstiegspunkt ins Programm dient. Die Main-Methode „macht“ zwar was, gibt allerdings kein Ergebnis zurück.

# Häufige Probleme

## Einführung

Problemstellung

Lösung

## Methoden

Parameter

Beispiele

Methoden sind Funktionen

Wertebereiche

Umsetzung

void

## Klassenvariablen

**Probleme**

Beispiel

Deklaration

## Objektorientierung

## Literatur

## Konventionen

- Methoden können nur einen Wert zurückgeben
- Methoden sollen auf gemeinsame Daten zugreifen (wird in den Übungen gebraucht!)

Lösung

⇒ Klassenvariablen

# Beispiel

## Einführung

- Problemstellung
- Lösung

## Methoden

- Parameter
- Beispiele
- Methoden sind Funktionen
- Wertebereiche
- Umsetzung
- void

## Klassenvariablen

- Probleme
- Beispiel**
- Deklaration

## Objektorientierung

## Literatur

## Konventionen

```
public class Zaehler {  
    public static int zaehlerStand = 0;  
  
    public static void Erhoehe(){  
        zaehlerStand++;  
    }  
    public static void Erniedrige(){  
        zaehlerStand--;  
    }  
}
```

Beide Methoden greifen auf den Zaehlerstand zu.

# Deklaration von Klassenvariablen

## Aufbau

```
public class Klasse{  
    //die Klassenvariable wird nach der Klasse  
    //nach der Klasse definiert  
    public static Typ name;  
  
    public static void Main(){...}  
}
```

## ABER

Statische Klassenvariablen mit Bedacht verwenden.  
Es kann sehr schnell unübersichtlich werden oder zu Seiteneffekten führen, wenn mehrere Methoden dieselbe Variable benutzen.

Einführung

Problemstellung

Lösung

Methoden

Parameter

Beispiele

Methoden sind Funktionen

Wertebereiche

Umsetzung

void

Klassenvariablen

Probleme

Beispiel

Deklaration

Objektorientierung

Literatur

Konventionen



## Einführung

Problemstellung

Lösung

## Methoden

Parameter

Beispiele

Methoden sind Funktionen

Wertebereiche

Umsetzung

void

## Klassenvariablen

Probleme

Beispiel

Deklaration

## Objektorientierung

## Literatur

## Konventionen

**Objektorientierung: eine kleine Einführung**  
Essentiell im Informatik-Studium  
(für die Übungen nicht so wichtig).

# Objektorientierung

## Einführung

- Problemstellung
- Lösung

## Methoden

- Parameter
- Beispiele
- Methoden sind Funktionen
- Wertebereiche
- Umsetzung
- void

## Klassenvariablen

- Probleme
- Beispiel
- Deklaration

## Objektorientierung

- Literatur

- Konventionen

### Frage

Was wollen wir in der Programmierung häufig erreichen?

Antwort: Die reale Welt modellieren.

Die Welt besteht aus Objekten.

### Objekte

Objekte haben Eigenschaften (Länge, Breite, Höhe, Farbe,...)  
Objekte haben Verhalten (leuchten, fahren, fliegen, reden, schlafen,...).

# Beispiel

## Einführung

Problemstellung

Lösung

## Methoden

Parameter

Beispiele

Methoden sind Funktionen

Wertebereiche

Umsetzung

void

## Klassenvariablen

Probleme

Beispiel

Deklaration

## Objektorientierung

## Literatur

## Konventionen

```
public class Main {  
  
    public static void Main() {  
        //Objekt anlegen  
        Auto auto = new Auto();  
  
        while (!auto.IsTankLeer()) {  
            auto.Fahre();  
        }  
    }  
}
```

# Beispiel

Die zugehörige Auto-Klasse.

```
public class Auto {  
    int tankInhalt = 100;  
  
    public boolean IsTankLeer(){  
        if(tankInhalt > 0){  
            return false;  
        }else{  
            return true;  
        }  
    }  
  
    public void Fahre(){  
        tankInhalt = tankInhalt - 1;  
    }  
}
```

Klasse anlegen: rechte Maustaste aufs Projekt  $\Rightarrow$  neue Klasse

Einführung

Problemstellung

Lösung

Methoden

Parameter

Beispiele

Methoden sind Funktionen

Wertebereiche

Umsetzung

void

Klassenvariablen

Probleme

Beispiel

Deklaration

Objektorientierung

Literatur

Konventionen

# Literatur

## Einführung

Problemstellung

Lösung

## Methoden

Parameter

Beispiele

Methoden sind Funktionen

Wertebereiche

Umsetzung

void

## Klassenvariablen

Probleme

Beispiel

Deklaration

## Objektorientierung

## Literatur

Konventionen

### Java



Christian Ullenboom: „Java ist auch eine Insel“

<http://openbook.galileocomputing.de/javainsel/>

### C#



Golo Roden: „Auf der Fährte von C#“

<http://www.guidetocsharp.de/>

# Kleiner Nachtrag zu Programmierkonventionen

## Java

Methoden beginnen klein, alle nachfolgenden Teilworte werden großgeschrieben.

⇒ `erhoeheZaehler()`

Genauso bei Variablennamen. ⇒ `int zaehlerStand;`

## C#

Methoden beginnen groß, alle nachfolgenden Teilworte werden großgeschrieben.

⇒ `ErhoeheZaehler()`

Variablennamen beginnen klein, alle nachfolgenden Teilworte werden großgeschrieben.

⇒ `int zaehlerStand;`

Wenn man sich nicht daran hält ist es kein Fehler, sieht nur manchmal unschön aus.

Einführung

Problemstellung

Lösung

Methoden

Parameter

Beispiele

Methoden sind Funktionen

Wertebereiche

Umsetzung

void

Klassenvariablen

Probleme

Beispiel

Deklaration

Objektorientierung

Literatur

Konventionen

# ENDE

## Einführung

Problemstellung

Lösung

## Methoden

Parameter

Beispiele

Methoden sind Funktionen

Wertebereiche

Umsetzung

void

## Klassenvariablen

Probleme

Beispiel

Deklaration

## Objektorientierung

## Literatur

## Konventionen

Viel Spaß bei der Übung.  
Evaluieren nicht vergessen!