3. Vortrag Methoden

3. Vortrag

Timo Lausen

Elektrotechnik
4. Semester

(in Regelzeit, jiiieehaa XD)

3. Vortrag Methoden

- Wiederholung
- Methoden
- Über Variablen
- Testen
- Debuggen

1. Wiederholung

Wiederholung

- Variablen und Zuweisungen

```
1   Typ name = wert;
```

Wiederholung

- Variablen und Zuweisungen

```
1   Typ name = wert;
```

```
2   //z.B.
3
4   int zahl = 10;
5   String text = “Hallo Welt!“;
6
7
8
9
10
```

Wiederholung

- Variablen und Zuweisungen

```
1   Typ name = wert;
```

```
2   //z.B.
3
4   int zahl = 10;
5   String text = “Hallo Welt!“;
6
7   int paul;
8   paul = 23;
9
10
```

Wiederholung

- Variablen und Zuweisungen

```
1   Typ name = wert;
```

```
2   //z.B.
3
4   int zahl = 10;
5   String text = “Hallo Welt!“;
6
7   int paul;
8   paul = 23;
9
10  paul = zahl;
```

Wiederholung

- Verzweigungen

```
1   if (bedingung) {
2
3   }
```

Wiederholung

- Verzweigungen

```
1   if (bedingung) {
2
3   }
```

```
4   if (sonneScheint) {
5      // Anweisung 1
6
7   } else if (regenFaellt) {
8      // Anweisung 2
9
10  } else {
11     // Anweisung 3
12  }
```

Wiederholung

- Schleifen

```
1   for (int i = start; i <= ende; i++) {
2      //Anweisung
3   }
```

Wiederholung

- Schleifen

```
1   for (int i = start; i <= ende; i++) {
2      //Anweisung
3   }
```

```
4   while (bedingung) {
5      //Anweisung
6   }
```

Wiederholung

- Arrays

```
1   int[] array = new int[laenge];
2
3   array[0] = 1;
4   array[1] = 2;
5   // u.s.w.
6   array[laenge - 1] = .....
```

```
7   int[] array = {1, 2, 3, 4, 5, 6, 7};
```

Wiederholung

- Mehrdimensionale Arrays

```
1   int[][] matrix = new int[3][3];
2
3   matrix[0][0] = 0;
4   matrix[1][0] = 1;
5   matrix[2][0] = 2;
6   //u.s.w.
7   matrix[2][2] = 9;
```

```
8   int[][] matrix = { {1,2,3},{4,5,6},{7,8,9} };
```

Wiederholung

Kurze Fragen?

2. Methoden

Methoden

Was ist eine Methode?

Methoden

Wörterbuch

Befehl: **Name**

Bedeutet übersetzt:

Anweisung1;
Anweisung2;
...

Methoden

Wörterbuch

Befehl: **Name**

Bedeutet übersetzt:

Anweisung1;
Anweisung2;
...

Beispiel:

- System.out.println()
- Math.random()

Methoden

Wie sieht eine Methode aus?

Methoden

Kopf:

- Name
- Parameter
- Rückgabetyp
- und noch anderes Zeug

Rumpf:

- Definition des Befehls

Methoden

```
public static  typ  name (typ Parameter1 , typ Parameter2 , …) {

        //hier kommt der Rumpf hin

}
```

Methoden

Rückgabetyp

Name der Methode

Parameter (Eingabewerte)

```
public static  typ  name (typ Parameter1 , typ Parameter2 , …) {

      //hier kommt der Rumpf hin


}
```

Methoden

Rückgabetyp

Name der Methode

Parameter (Eingabewerte)

```
public static  typ  name (typ Parameter1 , typ Parameter2 , …) {

    //hier kommt der Rumpf hin

}
```

Mögliche Datentypen:

<u>Parameter: (beliebig viele)</u>
Einfache Typen (z.B. int, double, boolean)
Komplexe Typen (z.B. Arrays, Objekte)
Keine Parameter ()

<u>Rückgabetypen: (nur einen)</u>
Einfache Typen
Komplexe Typen
void (gar kein Typ)

Methoden

Erster eigener Befehl

Namensschilder für Tutoren

Methoden

```
 1   public class NamensSchild {
 2
 3
 4
 5
 6
 7
 8
 9
10
11   public static void main (String[] args) {
12
13
14
15   }
16
17 }
```

Methoden

```
 1   public class NamensSchild {
 2
 3   public static void schild() {
 4
 5
 6
 7
 8
 9    }
10
11   public static void main (String[] args) {
12
13
14
15   }
16
17 }
```

Methoden

```
 1   public class NamensSchild {
 2
 3   public static void schild() {
 4
 5        System.out.println("##################");
 6        System.out.println("# Timo Lausen 24 #");
 7        System.out.println("##################");
 8
 9    }
10
11   public static void main (String[] args) {
12
13
14
15   }
16
17 }
```

Methoden

```
 1   public class NamensSchild {
 2
 3   public static void schild() {
 4
 5        System.out.println("##################");
 6        System.out.println("# Timo Lausen 24 #");
 7        System.out.println("##################");
 8
 9    }
10
11   public static void main (String[] args) {
12
13        schild();  //führt den Befehl aus
14
15   }
16
17 }
```

Methoden

```
 1   public class NamensSchild {
 2
 3   public static void schild() {
 4
 5       System.out.println("##################");
 6       System.out.println("# Timo Lausen 24 #");
 7       System.out.println("##################");
 8
 9    }
10
11   public static void main (String[] args) {
12
13       schild();  //führt den Befehl aus
14
15   }
16
17 }
```

Konsole:

```
################
# Timo Lausen #
################
```

Methoden

Problem:

- Tutoren haben unterschiedliche Namen

Methoden

Problem:

- Tutoren haben unterschiedliche Namen

Lösung:

- Es werden nur noch Tutoren eingestellt die Timo Lausen heißen

Methoden

Problem:

- Tutoren haben unterschiedliche Namen

Lösung:

- die Methode wird um zusätzlich Parameter erweitert:

  Name des Tutors, Alter des Tutors

Methoden

```
 1   public class NamensSchild2 {
 2
 3   public static void superSchild (/*Platzhalter*/) {
 4
 5
 6
 7
 8
 9    }
10
11   public static void main (String[] args) {
12
13
14
14
15   }
16
17 }
```

Methoden

```
 1   public class NamensSchild2 {
 2
 3   public static void superSchild (String name, int alter) {
 4
 5
 6
 7
 8
 9    }
10
11   public static void main (String[] args) {
12
13
14
14
15   }
16
17 }
```

Methoden

```
 1   public class NamensSchild2 {
 2
 3   public static void superSchild (String name, int alter) {
 4
 5       System.out.println("##################");
 6       System.out.println("# " + name + " " + alter + " #");
 7       System.out.println("##################");
 8
 9    }
10
11   public static void main (String[] args) {
12
13
14
14
15   }
16
17 }
```

Methoden

```
 1   public class NamensSchild2 {
 2
 3   public static void superSchild (String name, int alter) {
 4
 5       System.out.println("##################");
 6       System.out.println("# " + name + " " + alter + " #");
 7       System.out.println("##################");
 8
 9    }
10
11   public static void main (String[] args) {
12
13       superSchild(.........
14
14
15   }
16
17 }
```

Methoden

```
 1   public class NamensSchild2 {
 2
 3   public static void superSchild (String name, int alter) {
 4
 5       System.out.println("##################");
 6       System.out.println("# " + name + " " + alter + " #");
 7       System.out.println("##################");
 8
 9    }
10
11   public static void main (String[] args) {
12
13       superSchild("Timo Lausen",.....
14
14
15   }
16
17 }
```

Methoden

```
 1   public class NamensSchild2 {
 2
 3   public static void superSchild (String name, int alter) {
 4
 5       System.out.println("##################");
 6       System.out.println("# " + name + " " + alter + " #");
 7       System.out.println("##################");
 8
 9    }
10
11   public static void main (String[] args) {
12
13       superSchild("Timo Lausen", 24);
14
14
15   }
16
17 }
```

Methoden

```
 1   public class NamensSchild2 {
 2
 3   public static void superSchild (String name, int alter) {
 4
 5       System.out.println("##################");
 6       System.out.println("# " + name + " " + alter + " #");
 7       System.out.println("##################");
 8
 9    }
10
11   public static void main (String[] args) {
12
13       superSchild("Timo Lausen", 24);
14       superSchild("Paul Herman", 23);
14
15   }
16
17 }
```

Methoden

```
 1   public class NamensSchild2 {
 2
 3   public static void superSchild (String name, int alter) {
 4
 5       System.out.println("##################");
 6       System.out.println("# " + name + " " + alter + " #");
 7       System.out.println("##################");
 8
 9    }
10
11   public static void main (String[] args) {
12
13       superSchild("Timo Lausen", 24);
14       superSchild("Paul Herman", 23);
14
15   }
16
17 }
```

Konsole:

```
####################
# Timo Lausen 24 #
####################

####################
# Paul Herman 23 #
####################
```

Methoden

Fragen?

Methoden

Beispiele

- System.out.println();
- Math.random();

Methoden

Beispiele

- System.out.println();
- Math.random();

Gibt es hier einen Unterschied?

Methoden

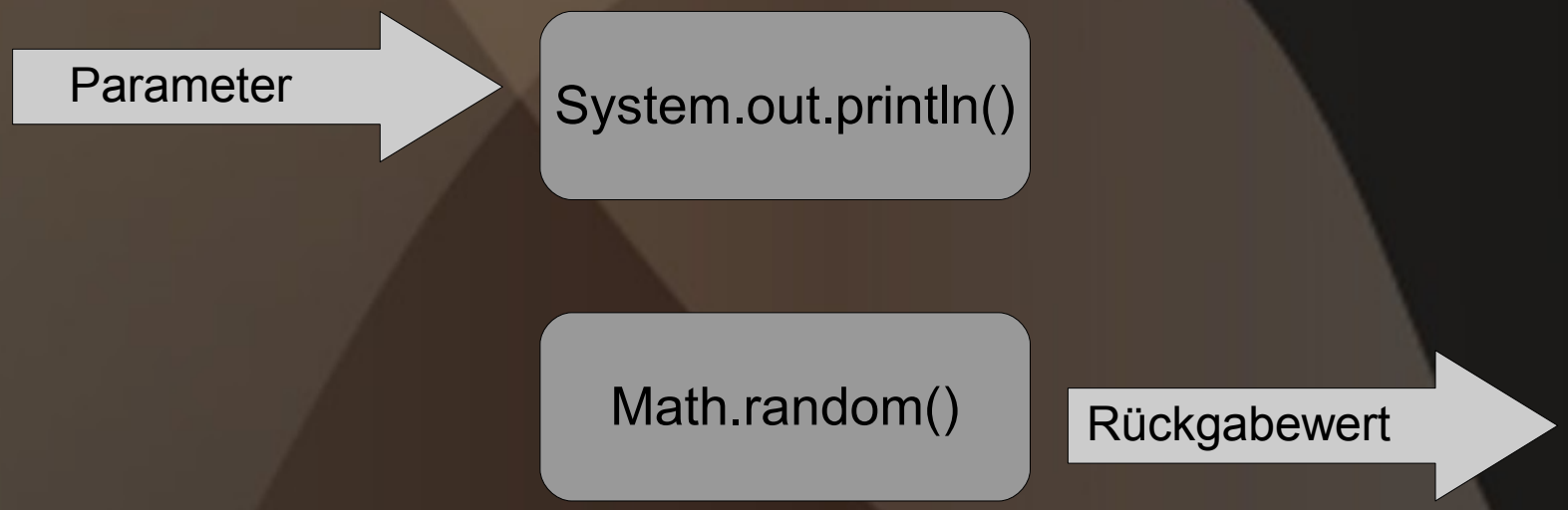
Beispiele

- System.out.println();
- Math.random();

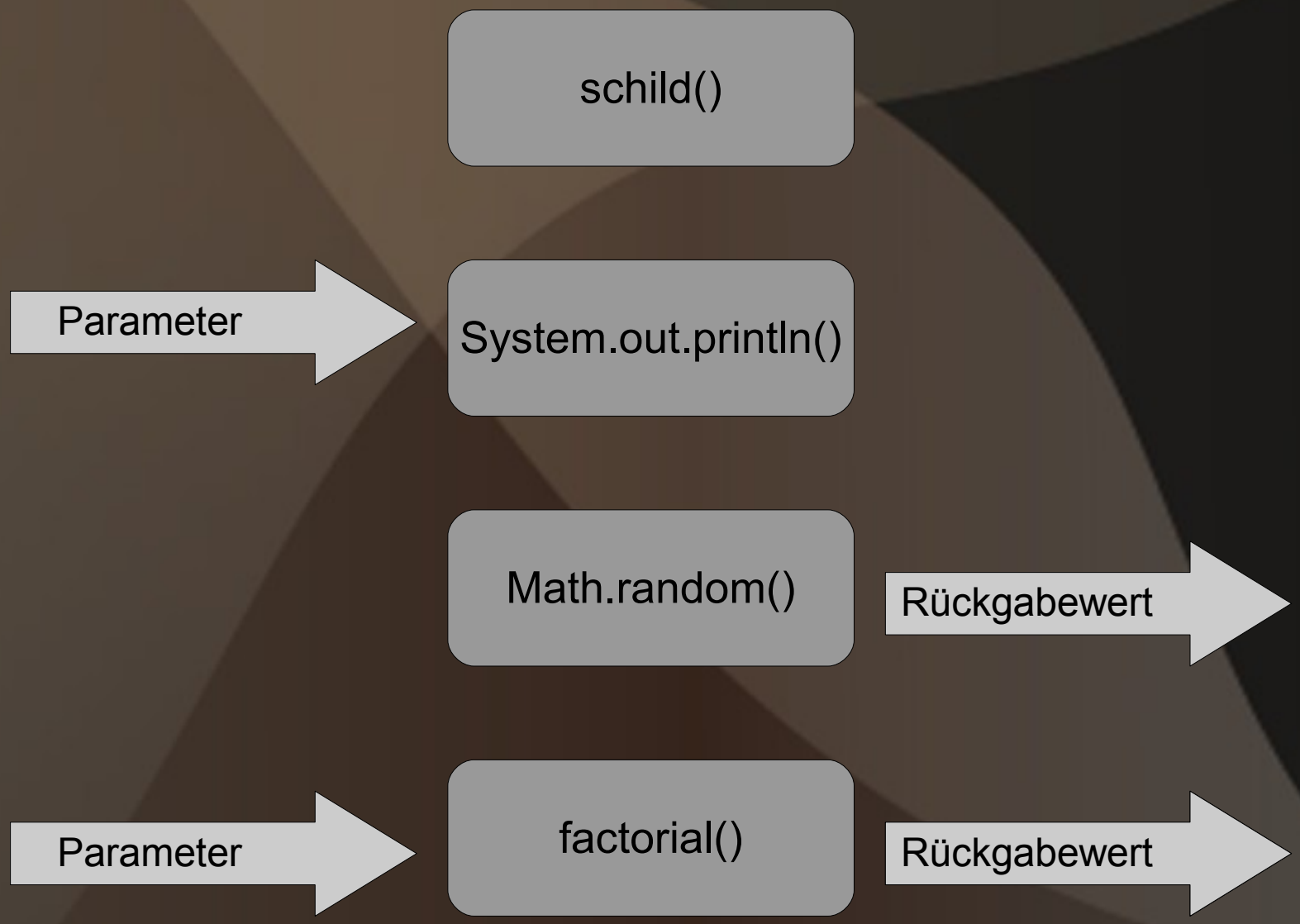
Gibt es hier einen Unterschied?

- System.out.println() schreib auf Konsole (void)
- Math.random() gibt eine Zahl zurück (typ)

Methoden



Methoden


schild()
Parameter
System.out.println()
Math.random()
Rückgabewert
Parameter
factorial()
Rückgabewert

Methoden

Die Fakultät der Zahl n soll berechnet werden

$f(n) = n!$ z.B. $f(4) = 1 \cdot 2 \cdot 3 \cdot 4 = 24$

Methoden

mathematische Funktion

Definition (Methodenkopf):

- Name: factorial
- Parameter: zahl
- Rückgabetyp: int

Methoden

```
 1  public class MyMath {
 2 //-------------------------------------------------------------------------------------
 3   public static /*Platzhalter*/ factorial (/*Platzhalter*/) {
 4
 5
 6
 7
 8
 9
10
11
12
13   }
14 //-------------------------------------------------------------------------------------
15   public static void main (String[] args) {
16
17
18
19
20
21
22
23   } }
```

Methoden

```
 1  public class MyMath {
 2 //-------------------------------------------------------------------------------------
 3   public static /*Platzhalter*/ factorial (int zahl) {
 4
 5
 6
 7
 8
 9
10
11
12
13   }
14 //-------------------------------------------------------------------------------------
15   public static void main (String[] args) {
16
17
18
19
20
21
22
23   } }
```

Methoden

```
 1  public class MyMath {
 2 //-------------------------------------------------------------------------------------
 3   public static int factorial (int zahl) {
 4
 5     int result = 1;
 6
 7
 8
 9
10
11
12      return result;
13   }
14 //-------------------------------------------------------------------------------------
15   public static void main (String[] args) {
16
17
18
19
20
21
22
23   } }
```

Methoden

```
 1  public class MyMath {
 2 //-------------------------------------------------------------------------------------
 3   public static int factorial (int zahl) {
 4
 5     int result = 1;
 6
 7      while(zahl > 0) {
 8         result = result * zahl;
 9         zahl = zahl - 1;
10      }
11
12      return result;
13   }
14 //-------------------------------------------------------------------------------------
15   public static void main (String[] args) {
16
17
18
19
20
21
22
23   } }
```

Methoden

```
 1  public class MyMath {
 2 //-------------------------------------------------------------------------------------
 3   public static int factorial (int zahl) {
 4
 5      int result = 1;
 6
 7       while(zahl > 0) {
 8          result = result * zahl;
 9          zahl = zahl - 1;
10       }
11
12       return result;
13    }
14 //-------------------------------------------------------------------------------------
15    public static void main (String[] args) {
16
17       int ergebnis = factorial(4);
18
19
20
21
22
23    } }
```

Methoden

```
 1  public class MyMath {
 2 //-------------------------------------------------------------------------------------
 3   public static int factorial (int zahl) {
 4
 5      int result = 1;
 6
 7       while(zahl > 0) {
 8          result = result * zahl;
 9          zahl = zahl - 1;
10       }
11
12       return result;
13    }
14 //-------------------------------------------------------------------------------------
15    public static void main (String[] args) {
16
17       int ergebnis = factorial(4);
18
19       int n = 3;
20       ergebnis = factorial(n);
21
22
23    } }
```

Methoden

```
 1  public class MyMath {
 2 //-------------------------------------------------------------------------------------
 3   public static int factorial (int zahl) {
 4
 5      int result = 1;
 6
 7       while(zahl > 0) {
 8          result = result * zahl;
 9          zahl = zahl - 1;
10       }
11
12       return result;
13    }
14 //-------------------------------------------------------------------------------------
15    public static void main (String[] args) {
16
17       int ergebnis = factorial(4);
18
19       int n = 3;
20       ergebnis = factorial(n);
21
22       System.out.println (“4! = “ + factorial(4));
23    } }
```

Methoden

```
 1  public class MyMath {
 2 //-------------------------------------------------------------------------------------
 3   public static int factorial (int zahl) {
 4
 5     int result = 1;
 6
 7      while(zahl > 0) {
 8         result = result * zahl;
 9         zahl = zahl - 1;
10      }
11
12      return result;
13   }
14 //-------------------------------------------------------------------------------------
15   public static void main (String[] args) {
16
17      int ergebnis = factorial(4);
18
19      int n = 3;
20      ergebnis = factorial(n);
21
22      System.out.println (“4! = “ + factorial(4));
23   } }
```

Konsole:

```
4!  =  24
```

Methoden

Methode:

```
public static typ methode (typ parameter1,...) {

}
```

Anwendung (z.B.):

```
methode();
a = methode(zahl1, zahl2);
if (methode(5) == true) {.....}
```

Methoden

Sinn von Methoden:

- Quellcode kann universell genutzt werden (z.B. superSchild())

- “auslagern“ von Code erhöht die Übersichtlichkeit

  (z.B. System.out.println())

Methoden

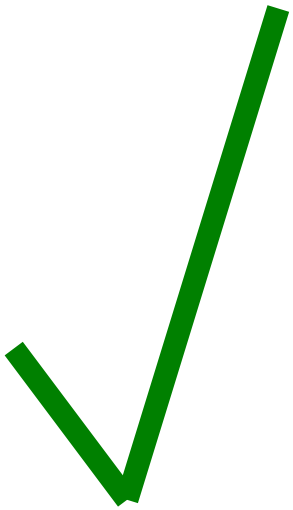
Fragen?

3. Variablen in Methoden

Variablen in Methoden

Variablen sind generell nicht überall verfügbar

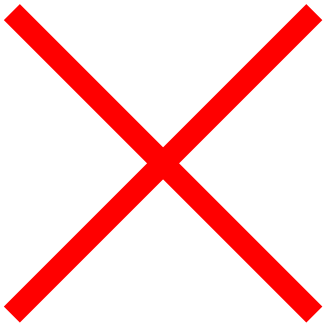
```
 1 public class MyMath{
 2
 3    public static int addiere (int zahl1, int zahl2) {
 4       int result = zahl1 + zahl2;
 5       .....
 6    }
 7
 8    public static int factorial (int zahl1) {
 9       int result = 1;
10       ....
11    }
12
13    public static void main (String[] args) {
14       int zahl1 = 5;
15       int zahl2 = 3;
16       int result = addiere(zahl1, zahl2);
17       .....
18    }
19 }
```



Variablen in Methoden

Variablen sind generell nicht überall verfügbar

```
 1 public class MyMath{
 2
 3    public static int addiere (int zahl1, int zahl2) {
 4       int result = zahl1 + zahl2;
 5       .....
 6    }
 7
 8    public static int factorial (int zahl1) {
 9       int result = 1;
10       ....
11    }
12
13    public static void main (String[] args) {
14       int zahl1 = 5;
15       int zahl2 = 3;
16       result = addiere(zahl1, zahl2);
17       .....
18    }
19 }
```



Variablen in Methoden

Merke:

- Variablen sind ausschließlich in den Methoden verfügbar, in denen sie definiert wurden

  (sie sind local)

- Ausnahme: Klassenvariablen

Variablen in Methoden

Aber wie kommen nun die Parameter in die Methode?

Variablen in Methoden

Aber wie kommen nun die Parameter in die Methode?

Sie werden rein kopiert.

Variablen in Methoden

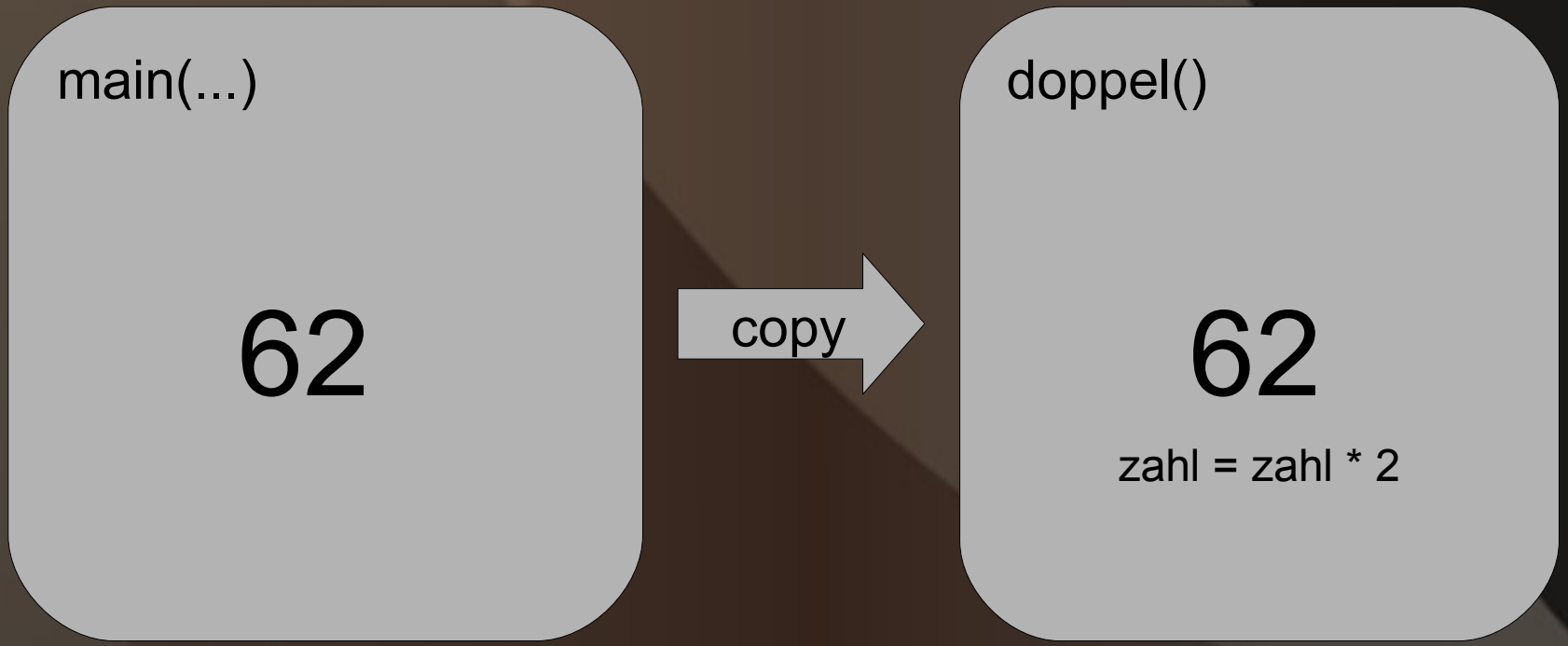
Call by Value:

public static int doppel (int zahl){...}

Variablen in Methoden

Call by Value:

public static int doppel (int zahl){...}



Variablen in Methoden

Call by Value:

public static int doppel (int zahl){...}

Variablen in Methoden

```
 1 public class MyMath {
 2
 3    public static int doppel (int value) {
 4       value = value * 2;
 5
 6       return value;
 7    }
 8
 9    public static void main (String[] args) {
10       int zahl = 62;
11
12       doppel(zahl);
13
14    }
15 }
```

Variablen in Methoden

```
 1 public class MyMath {
 2
 3    public static int doppel (int value) {
 4       value = value * 2;
 5       System.out.println("Zahl während Ausführung von doppel: " + value);
 6       return value;
 7    }
 8
 9    public static void main (String[] args) {
10       int zahl = 62;
11       System.out.println("Zahl vor Ausführung von doppel: " + zahl);
12       doppel(zahl);
13       System.out.println("Zahl nach Ausführung von doppel: " + zahl);
14    }
15 }
```

Variablen in Methoden

```
 1 public class MyMath {
 2
 3    public static int doppel (int value) {
 4       value = value * 2;
 5       System.out.println("Zahl während Ausführung von doppel: " + value);
 6       return value;
 7    }
 8
 9    public static void main (String[] args) {
10       int zahl = 62;
11       System.out.println("Zahl vor Ausführung von doppel: " + zahl);
12       doppel(zahl);
13       System.out.println("Zahl nach Ausführung von doppel: " + zahl);
14    }
15 }
```

Konsole:

```
Zahl vor Ausführung von doppel: 62
Zahl während Ausführung von doppel: 124
Zahl nach Ausführung von doppel: 62
```

Variablen in Methoden

Problem:

Bei sehr viel Daten zeit- und speicheraufwendig

Variablen in Methoden

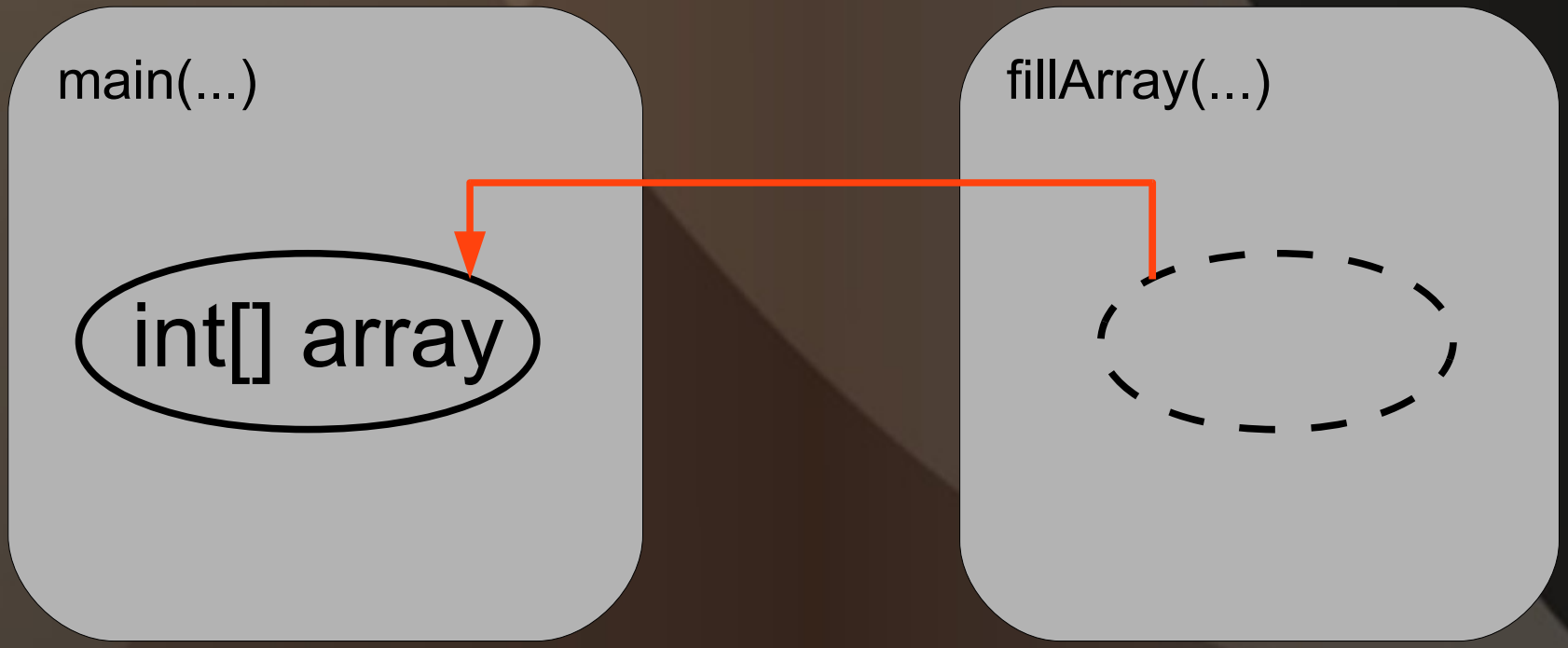
Call by Reference

public static void fillArray (int[] array){...}

Variablen in Methoden

Call by Reference

public static void fillArray (int[] array){...}



Variablen in Methoden

```
 1 public class MyArray{
 2
 3    public static void fillArray (int[] array) {
 4       for (int i = 0; i < array.length; i++) {
 5          array[i] = i;
 6       }
 7    }
 8
 9    public static void main (String[] args) {
10
11       int[] array = new int[10000];
12
13       System.out.println("Element 3728 vor Befuellen: " + array[3728]);
14       fillArray(array);
15       System.out.println("Element 3728 nach Befuellen: " + array[3728]);
16    }
17 }
```

Variablen in Methoden

```
 1 public class MyArray{
 2
 3    public static void fillArray (int[] array) {
 4       for (int i = 0; i < array.length; i++) {
 5          array[i] = i;
 6       }
 7    }
 8
 9    public static void main (String[] args) {
10
11       int[] array = new int[10000];
12
13       System.out.println("Element 3728 vor Befuellen: " + array[3728]);
14       fillArray(array);
15       System.out.println("Element 3728 nach Befuellen: " + array[3728]);
16    }
17 }
```

Konsole

```
Element 3728 vor Befuellen: 0
Element 3728 nach Befuellen:3728
```

Variablen in Methoden

Wann und wie verwendet man call by value/reference?

Variablen in Methoden

Call by value:	Call by reference
einfache Datentypen	komplexe Datentypen
(int, boolean, double,...)	(Arrays, Objekte,...)

Wie: ganz automatisch (geht gar nicht anders)

Variablen in Methoden

Fragen?

4. Testen

Testen

Was ist Testen?

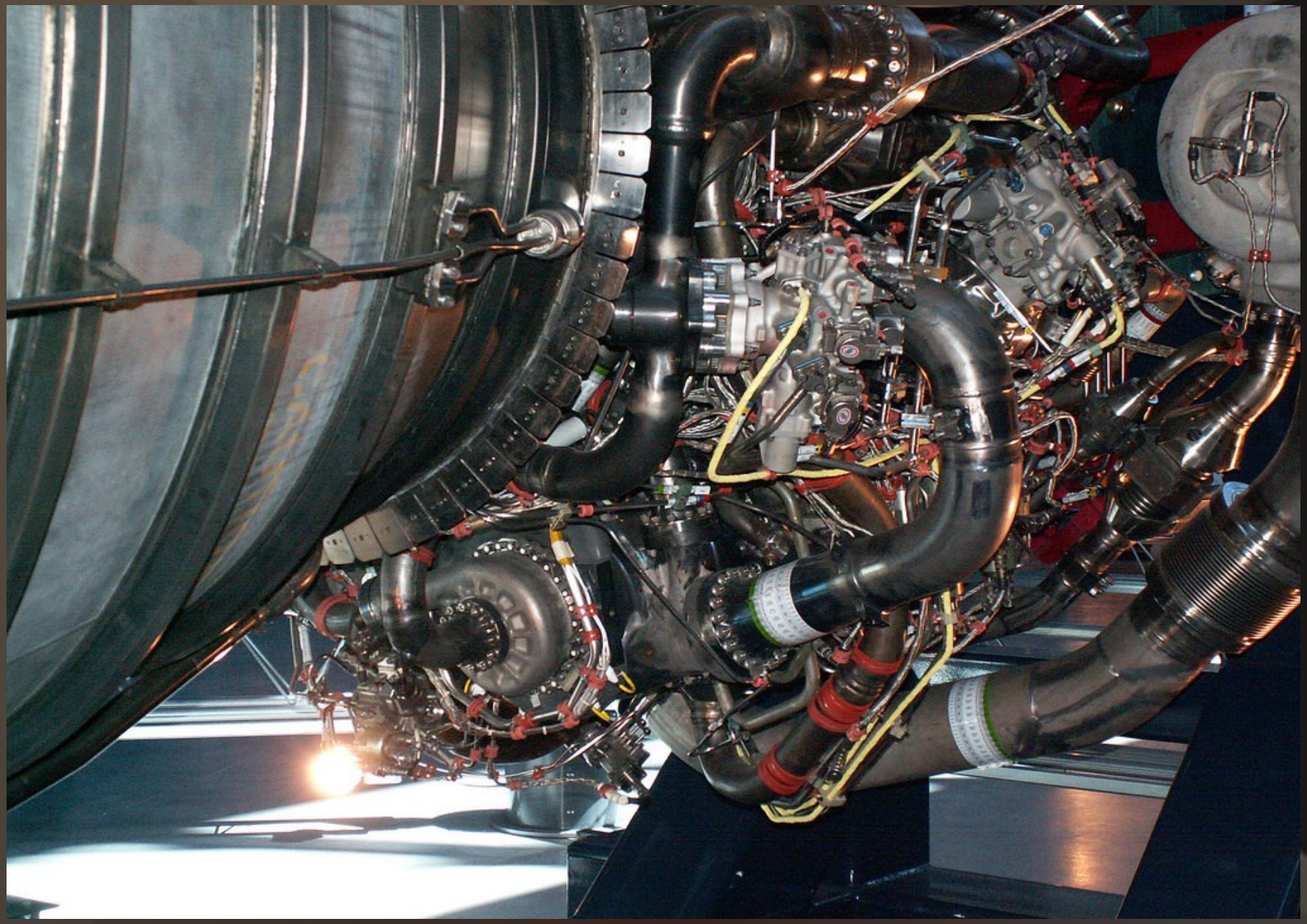
Testen

Und was bringt mir das?

Testen

Testen

Testen



Testen

Testen

Wie wird getestet?

Testen

Wie wird getestet?

Mit einem Testprogramm

Testen

Vor dem Schreiben der Methode:

- Was soll die Methode machen?
- Welche Probleme könnten auftreten?
  (z.B. Division durch 0)
- Fehler durch Fehlbedienung möglich?

Muss alles mit Test überprüft werden!

Testen

```
 1   public class MyMath {
 2  //-----------------------------------------------------------------------------------------
 3    public static int factorial (int zahl) {
 4
 5       int result = 1;
 6
 7        while(zahl > 0) {
 8           result = result * zahl;
 9           zahl = zahl - 1;
10        }
11
12        return result;
13    }
14  //-----------------------------------------------------------------------------------------
15    public static void main (String[] args) {
16
17        System.out.println("4!  Erwartet: 24; Ergebnis: " + factorial(4));
18        System.out.println("1!  Erwartet:   1; Ergebnis: " + factorial(1));
19        System.out.println("0!  Erwartet:   1; Ergebnis: " + factorial(0));
20        System.out.println("(-1)!  Erwartet: nicht definiert; Ergebnis: " + factorial(-1));
21
22    }
23 }
```

Testen

```
 1  public class MyMath {
 2 //-----------------------------------------------------------------------------------------
 3   public static int factorial (int zahl) {
 4
 5      int result = 1;
 6
 7       while(zahl > 0) {
 8          result = result * zahl;
 9          zahl = zahl - 1;
10       }
11
12       return result;
13   }
14 //
```

Konsole:

```
4!  Erwartet: 24; Ergebnis: 24
1!  Erwartet:   1; Ergebnis: 1
0!  Erwartet:   1; Ergebnis: 1
(-1)!  Erwartet: nicht definiert; Ergebnis: 1
```

Testen

<u>Wichtige Regeln:</u>

- Übliche Fälle, Sonderfälle und Randwerte testen
- Blöde Fälle testen (besonders wichtig)
  (z.B. -1!, ln(-23), 1/0, etc.)
- Immer mit der Blödheit des Benutzers rechnen
  (b.z.w. mit der eigenen)

Testen

Fragen?

5. Debuggen

Debuggen

Wie finde ich die lose Schraube?

Debuggen

- Fehler eingrenzen
  (wo und wann tritt der Fehler auf ?)

- Welche Zwischenergebnisse sollten hier vorliegen?
  (mit System.out.println() jedes Ergebniss ausgeben)

- Parameter überprüfen / event. Folgefehler
  (factorial(n) ist nur 24, wenn n = 4)

Debuggen

```
 1 public class MyMath{
 2
 3   public static int modulo (int zahl, int divisor){
 4      int result = zahl;
 5
 6
 7      while (result > divisor) {
 8
 9         result = result – divisor;
10
11      }
12
13     return result;
14   }
15
16   public static void main (String[] args) {
17      System.out.println("6 % 2 = " + modulo(6,2));
18   }
19 }
```

Debuggen

```
 1 public class MyMath{
 2
 3   public static int modulo (int zahl, int divisor){
 4      int result = zahl;
 5
 6
 7      while (result > divisor) {
 8
 9         result = result – divisor;
10
11      }
12
13     return result;
14   }
15
16   public static void main (String[] args) {
17      System.out.println("6 % 2 = " + modulo(6,2)
18   }
19 }
```

Konsole:

```
6 % 2 = 2
```

Debuggen

```
 1 public class MyMath{
 2
 3   public static int modulo (int zahl, int divisor){
 4      int result = zahl;
 5
 6      System.out.println(zahl + “%“ + divisor);  //stimmen die Parameter?
 7      while (result > divisor) {
 8
 9         result = result – divisor;
10
11      }
12
13     return result;
14   }
15
16   public static void main (String[] args) {
17      System.out.println(“6 % 2 = “ + modulo(6,2));
18   }
19 }
```

Debuggen

```
 1 public class MyMath{
 2
 3   public static int modulo (int zahl, int divisor){
 4      int result = zahl;
 5
 6      System.out.println(zahl + “%“ + divisor);  //stimmen die Parameter?
 7      while (result > divisor) {
 8         System.out.print(result + “ - “ + divisor + “ = “); //Rechnung...
 9         result = result – divisor;
10         System.out.println(result);  //... und Ergebniss
11      }
12
13     return result;
14   }
15
16   public static void main (String[] args) {
17      System.out.println(“6 % 2 = “ + modulo(6,2));
18   }
19 }
```

Debuggen

```
 1 public class MyMath{
 2
 3   public static int modulo (int zahl, int divisor){
 4      int result = zahl;
 5
 6      System.out.println(zahl + “%“ + divisor);  //stimmen die Parameter?
 7      while (result > divisor) {
 8         System.out.print(result + “ - “ + divisor + “ = “); //Rechnung...
 9         result = result – divisor;
10         System.out.println(result);  //... und Ergebniss
11      }
12     System.out.println(“Schleifenabbruch“);
13     return result;
14   }
15
16   public static void main (String[] args) {
17      System.out.println(“6 % 2 = “ + modulo(6,2));
18   }
19 }
```

Debuggen

```
 1 public class MyMath{
 2
 3   public static int modulo (int zahl, int divisor){
 4      int result = zahl;
 5
 6      System.out.println(zahl + “%“ + divisor);  //stimmen die Parameter?
 7      while (result > divisor) {
 8         System.out.print(result + “ - “ + divisor + “ = “); //Rechnung...
 9         result = result – divisor;
10         System.out.println(result);  //... und Ergebniss
11      }
12     System.out.println(“Schleifenabbruch“);
13     return result;
14   }
15
16   public static void main (String[] args) {
17      System.out.println(“6 % 2 = “ + modulo(6,2)
18   }
19 }
```

Konsole:

```
6 % 2
6 - 2 = 4
4 - 2 = 2
Schleifenabbruch
```

Debuggen

```
 1 public class MyMath{
 2
 3   public static int modulo (int zahl, int divisor){
 4      int result = zahl;
 5
 6      //System.out.println(zahl + "%" + divisor);  //stimmen die Parameter
 7      while (result >= divisor) {
 8         //System.out.print(result + " - " + divisor + " = "); //Rechnung...
 9         result = result – divisor;
10         //System.out.println(result);  //... und Ergebniss
11      }
12     //System.out.println("Schleifenabbruch");
13     return result;
14   }
15
16   public static void main (String[] args) {
17      System.out.println("6 % 2 = " + modulo(6,2));
18   }
19 }
```

Konsole:

```
6 % 2 = 0
```

Debuggen

Fragen?

Und nun?

- Übung im TEL

empfolene Aufgaben:

  - Schaltjahr-Aufgabe (3)
  - lineare Funktionen (5)
  - Spaß mit Quersummen (5-7)

Und nun?

- Übung im TEL

empfolene Aufgaben:

- Schaltjahr-Aufgabe (3)
- lineare Funktionen (5)
- Spaß mit Quersummen (5-7)

Viel Spaß und guten Wirkungsgrad!