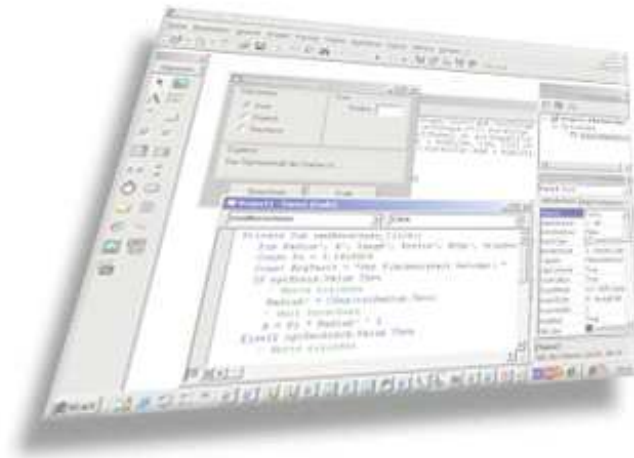
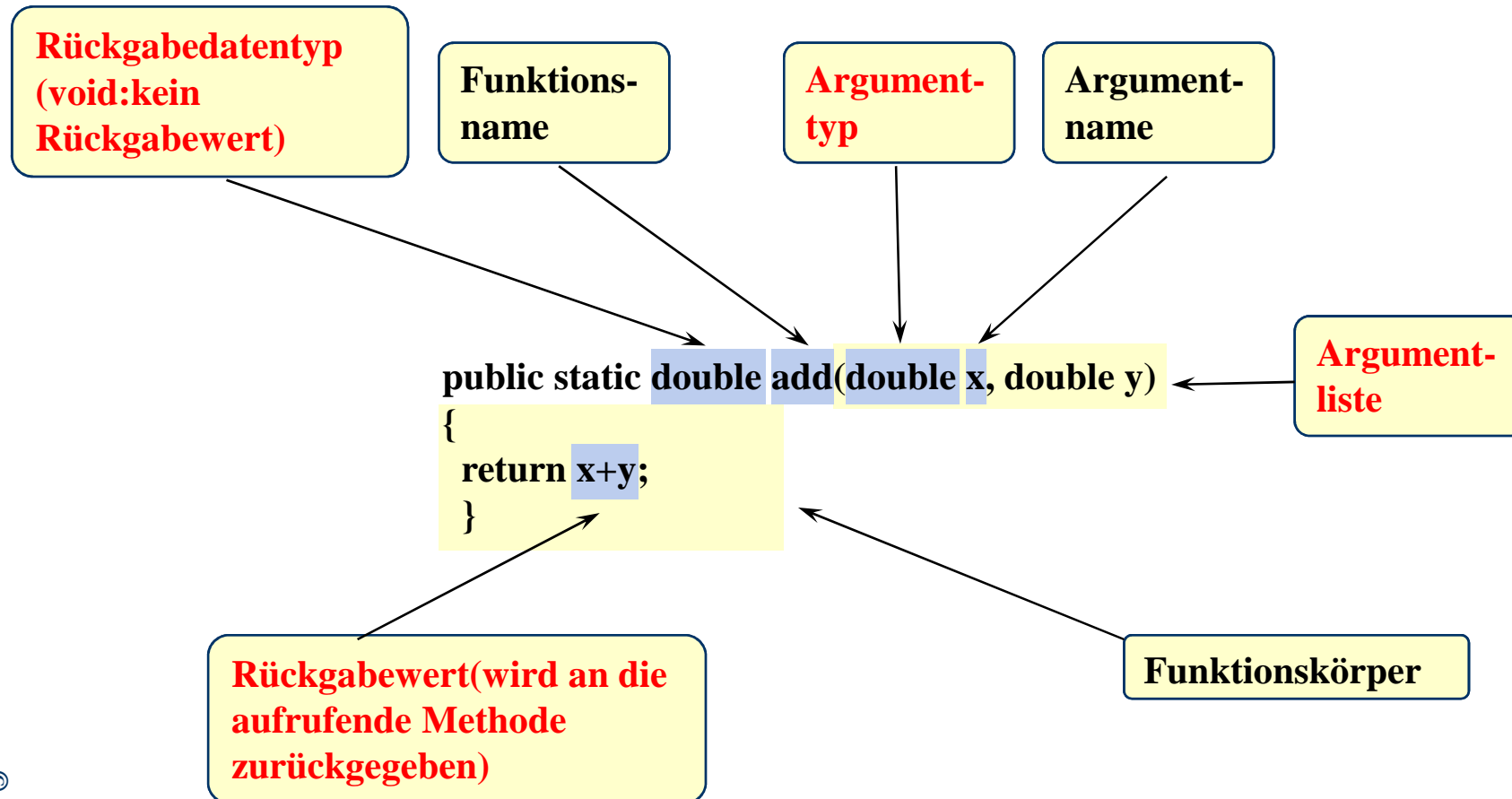


Methoden(Funktionen/Prozeduren)





Einführendes Beispiel I





Einführendes Beispiel II

```
public class Aufgaben {
```

```
    public static double add(double x, double y) {  
        return x+y;  
    }
```

```
    public static void main(String[] args) {  
        double zahl1, zahl2, summe;  
        zahl1=3;  
        zahl2=7;  
        summe=add(zahl1,zahl2);  
        System.out.println("Summe:"+summe);  
    }  
}
```

Jede Datei besteht(bis jetzt!)
aus einer Klasse

Die benutzerdefinierte
Methode und die
Hauptmethode sind in einem
Programm



Einführendes Beispiel III

Aufruf der Methode

```
public static void main(String[] args) {
    double zahl1, zahl2, summe;
    zahl1=3;
    zahl2=7;
    summe=add(zahl1,zahl2);
    System.out.println("Summe:"+summe);
}
```

```
public static double add(double x,double y){
    return x+y;
}
```

F7: Durchläuft jede Anweisung
F8: Überspringt die Methoden

Rückgabe des
 Funktionswertes und
Verlassen der Methode



Einführendes Beispiel IV

Zahl1 = 3 ; Zahl1 = 7

summe = add (zahl1,zahl2);

Anzahl der Argumente = Anzahl der Parameter

Übergeben der Werte an die Methode

```
public static double add ( double x,double y ){
    return x+y;
}
```

x = 3 ; y = 7

| | |
|---------------------|----------------|
| summe=add(1); | Compilerfehler |
| summe=add(1, 2); | OK |
| add(1, 2); | OK |
| summe=add(1,2,3); | Compilerfehler |
| summe=add(1,2,3,4); | Compilerfehler |



Einführendes Beispiel V

```
double summe, zahl1, zahl2; int x; bool y;
```

```
summe = add (zahl1, zahl2);
```

Datentypen (double, int, bool, string) müssen an jeder Position übereinstimmen!

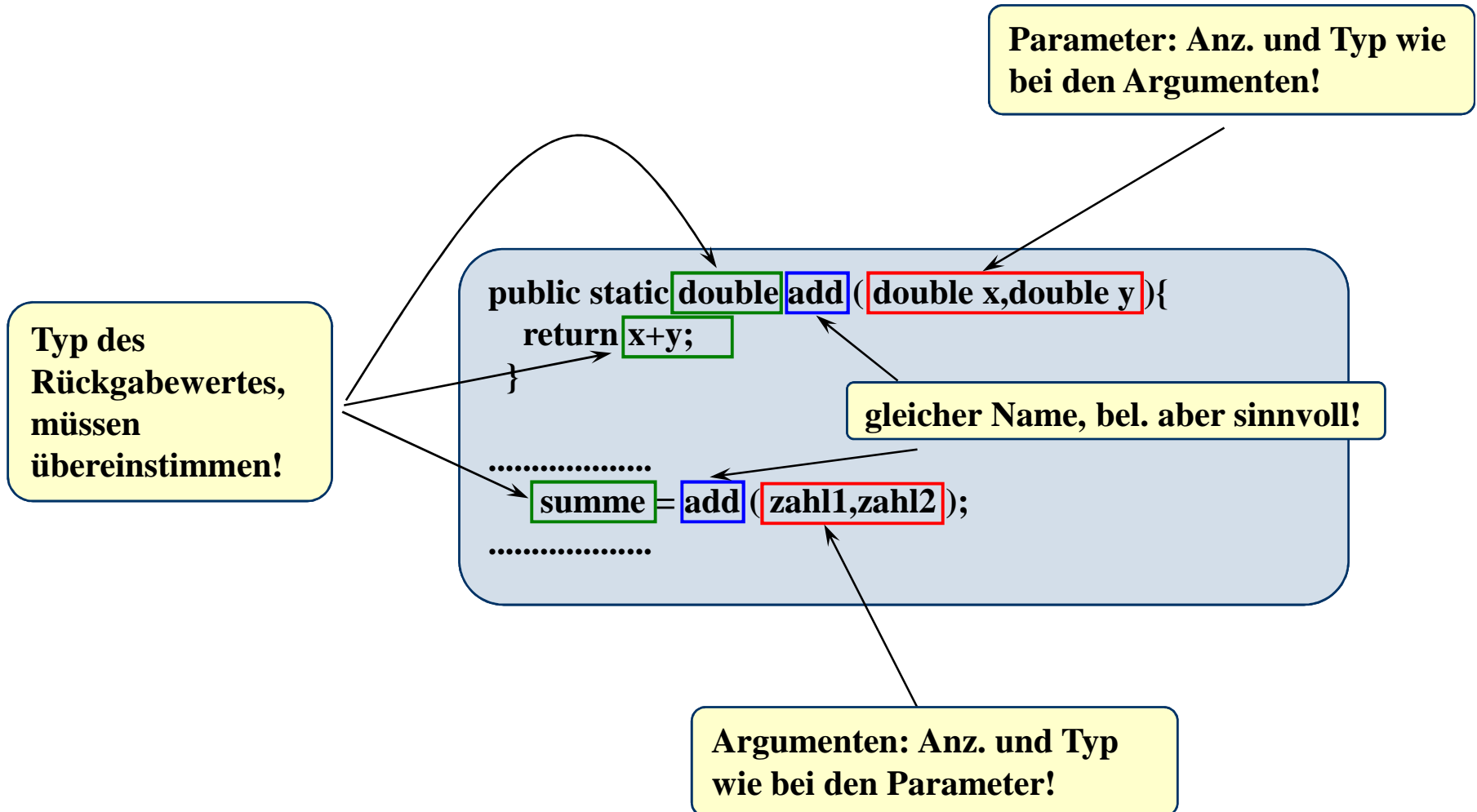
```
public static double add (double x, double y) {
    return x+y;
}
```

Auch der Datentyp des Rückgabewertes!

```
summe = add (x, y);
x = add (zahl1, zahl2);
zahl1 = add (zahl1, zahl2);
```



Einführendes Beispiel VI





Beispiel: Quadratische Funktion I

```
public class Parabel {  
  
    private static double f(double x){  
        return x*x;  
    }  
  
    public static void main(String[] args) {  
        double zahl, quadrat;  
        zahl=3;  
        quadrat=f(zahl);  
        System.out.println("Quadrat:"+quadrat);  
    }  
}
```




Beispiel: Quadratische Funktion II

```
public class Parabel {  
  
    private static double f(double x){  
        return x*x;  
    }  
  
    public static void main(String[] args) {  
        double zahl, q1,q2,q3;  
        zahl=3;  
        q1=f(zahl+2);  
        q2=f(-5);  
        q3=f(-2)+2*f(-4)*zahl;  
    }  
}
```



Beispiel: Quadratische Funktion III

```
public class QuadGl {  
  
    private static double xhoch2(double x){  
        return x*x;  
    }  
    private static double quadFunk(double x){  
        return 2*xhoch2(x)-4*x+4;  
    }  
    public static void main(String[] args) {  
        double res;  
        res=quadFunk(3);  
        System.out.println("Funktionswert:"+res);  
        System.out.println("Funktionswert:"+quadFunk(3));  
    }  
}
```



Beispiel: Quadratische Funktion IV

```
public class DasKlapptLeiderNicht{

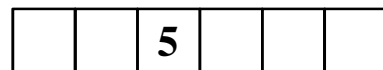
    private static int ÄndereArgument(int x){
        x=x*2;
        return 2*x;
    }

    public static void main(String[] args) {
        int a=3;
        int b=5;
        ÄndereArgument (b);
        System.out.println("a:"+a+" b:"+b);
    }
}
```

Kopie des Wertes
von einer
Speicherstelle zu
einer andern:
„Übergabe einer
Kopie des Wertes“:
„by value“

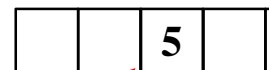
Unterschied in
anderen
Programmiersprach
en:
„by reference“
In Java gibt es das
NICHT!

Haupt-
speicher:



b

main



x

ÄndereArgument



Die Notwendigkeit eines „returns“

```
public static int max(int x,int y) {  
    if (x>y)  
        return x;  
    else  
        return y;  
}
```

Eine Methode **MUSS**
einen Rückgabewert
haben!

```
public static int max(int x,int y) {  
    if (x>y)  
        return x;  
    if (x<y)  
        return y;  
      
}
```

Hier „merkt“ der
Interpreter, dass
möglicherweise diese
Methode ohne return
durchlaufen wird!

```
test.java:8:3: error: missing return statement  
    }
```

Beispiel: Methoden ohne Rückgabewert

```
public class NoReturn {  
    private static void noRet(){  
        System.out.println("Kein Rueckgabewert");  
    }  
    public static void main(String[] args) {  
        noRet();  
    }  
}
```

void: Kein Rückgabewert!

„Normale“ Nomenklatur in anderen Programmiersprachen:
Ein Rückgabewert: Funktionen
Kein Rückgabewert: Prozeduren



Beispiel: Methoden mit boolean

```
public static boolean isGreater(int x,int y) {  
    return x>y;  
}
```

Boolean: true/false

```
public static void main(String[] args) {  
    boolean res;  
    res=isGreater(5,3);  
    if (res)  
        System.out.println("5 größer als 3!");  
    if (isGreater(5,3))  
        System.out.println("5 größer als 3!");  
}
```



Globale/Lokale Variablen Ia

```
public class Aufgaben {  
  
    static double result;  
    public static double add(double x,double y){  
        result=x+y;  
        return x+y;  
    }  
}
```

```
public static void main(String[] args) {  
    double zahl1, zahl2, summe;  
    zahl1=3;  
    zahl2=7;  
    summe=add(zahl1,zahl2);  
    System.out.println("Summe:"+summe);  
    System.out.println("Summe:"+result);  
}
```

lokale Variable

Gültigkeitsbereich



Globale/Lokale Variablen Ib

```
public class Aufgaben {
```

lokale Variable

```
    static double result;
```

```
    public static double add(double x, double y){  
        result=x+y;  
        return x+y;  
    }
```

Gültigkeitsbereich

```
    public static void main(String[] args) {
```

```
        double zahl1, zahl2,summe;
```

```
        zahl1=3;
```

```
        zahl2=7;
```

```
        summe=add(zahl1,zahl2);
```

```
        System.out.println("Summe:"+summe);
```

```
        System.out.println("Summe:"+result);
```

```
    }  
}
```




Globale/Lokale Variablen Ic

```
public class Aufgaben {
```

Globale Variable

```
    static double result;
```

```
    public static double add(double x, double y) {  
        result = x + y;  
        return x + y;  
    }
```

Gültigkeitsbereich

```
    public static void main(String[] args) {  
        double zahl1, zahl2, summe;  
        zahl1 = 3;  
        zahl2 = 7;  
        summe = add(zahl1, zahl2);  
        System.out.println("Summe:" + summe);  
        System.out.println("Summe:" + result);  
    }  
}
```