

Java Methoden

2021-09-29 Christian Ludl

Methoden

- Methodenname
 - Parameter
 - Rückgabewert
 - Körper (Rumpf)
-
- Deklaration
 - Aufruf

engl.: “declaration”
=
Erklärung, Ankündigung, Vereinbarung

z.B. “declaration of independence” = Unabhängigkeitserklärung

Methodendeklaration

- **WAS** braucht die Methode, um ihre Aufgabe erfüllen zu können?
- **WAS** ist das Ergebnis der Methode?
- Methoden dürfen in Java NICHT geschachtelt werden (keine Methoden innerhalb von Methoden deklarieren).
- Konvention: Methodennamen beginnen mit Kleinbuchstaben und werden in CamelCase geschrieben.

Methodenaufruf

- **WAS** muss ich der Methode übergeben?
- **WAS** bekomme ich von der Methode zurück?
- **WO** gehört der Methodenaufruf hin?
- **Parameter** werden **OHNE Datentyp** angegeben !!!

Deklaration <> Aufruf

Variablen

```
int x;  
String text;  
  
... ..  
  
// Aufruf:  
x = 42;  
text = "Don't panic!";
```

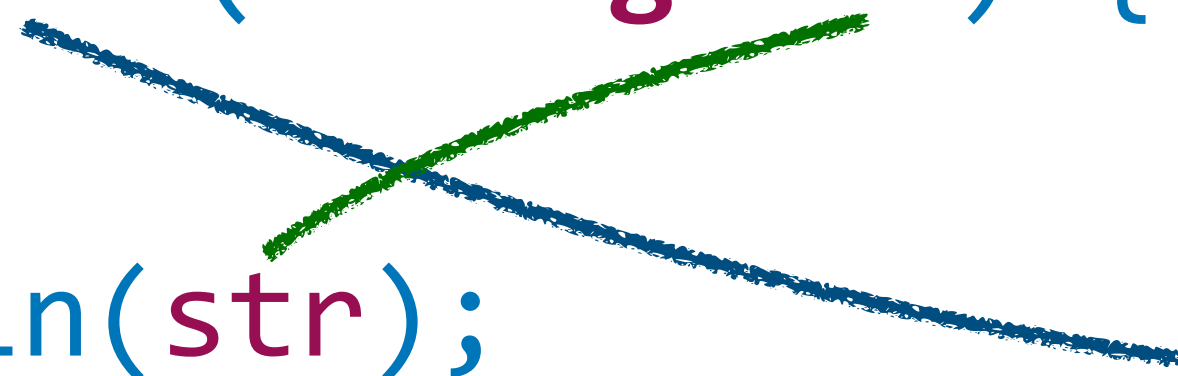
Methoden

```
public static int myCalc() {...}  
public static void printText(String myString) {...}  
  
... ..  
  
// Aufruf:  
result = myCalc();  
printText("Hello HTL-HL!");
```

Parameter

Methode

```
public static void methodName(String str) {  
    // ...  
    System.out.println(str);  
}
```



A blue arrow originates from the `text` parameter in the `methodName(text);` call on the right and points to the `str` parameter in the `methodName(String str)` definition on the left. A green arrow points from the `String` type in the definition to the `String` type in the call.

Aufruf

```
// ... (main)-Methode  
String text = "Hallo";  
methodName(text);
```

Parameter

Methode

```
public static void methodName(int i, float j)  
{  
    // ...  
}
```

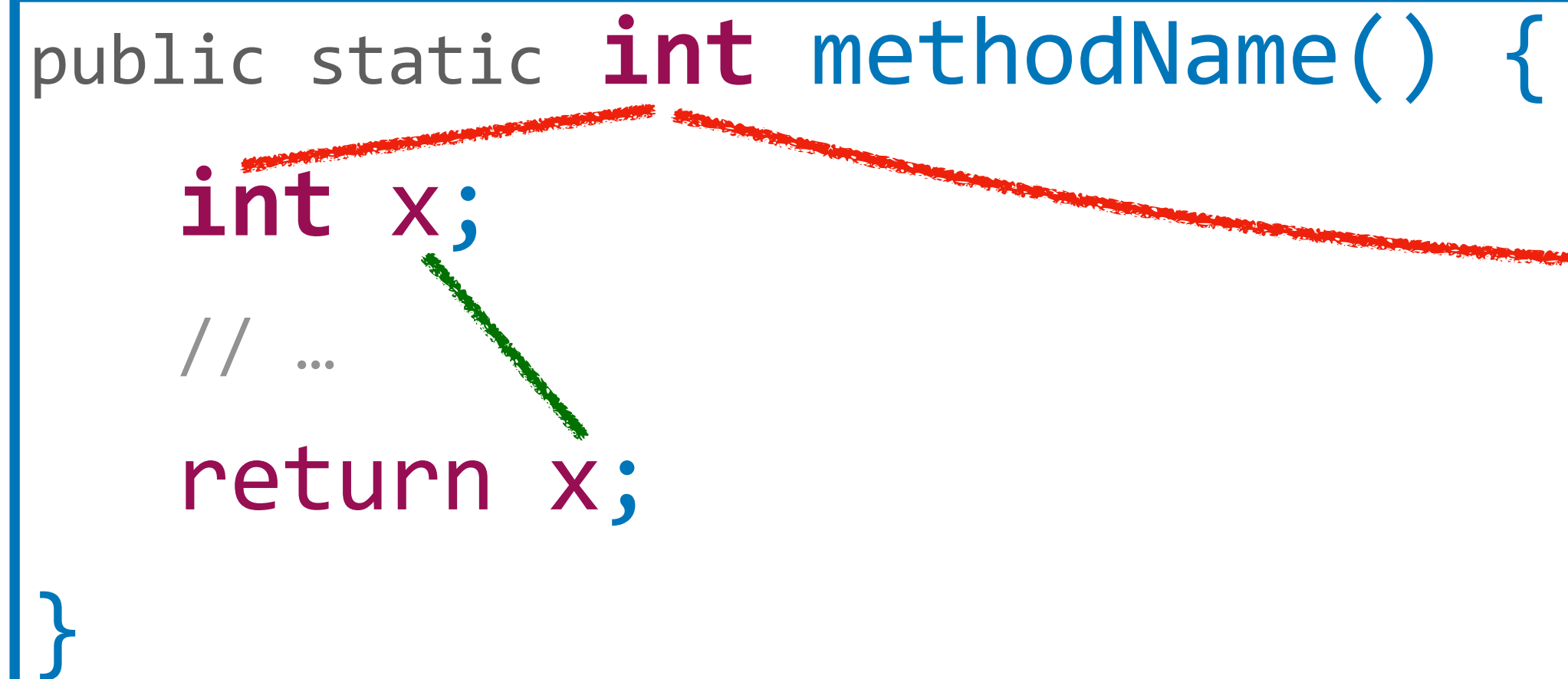
Aufruf

```
// ... (main)-Methode  
int i;  
float j;  
// ...  
i = 42;  
j = 3.14;  
methodName(i, j);
```


Rückgabe

Methode

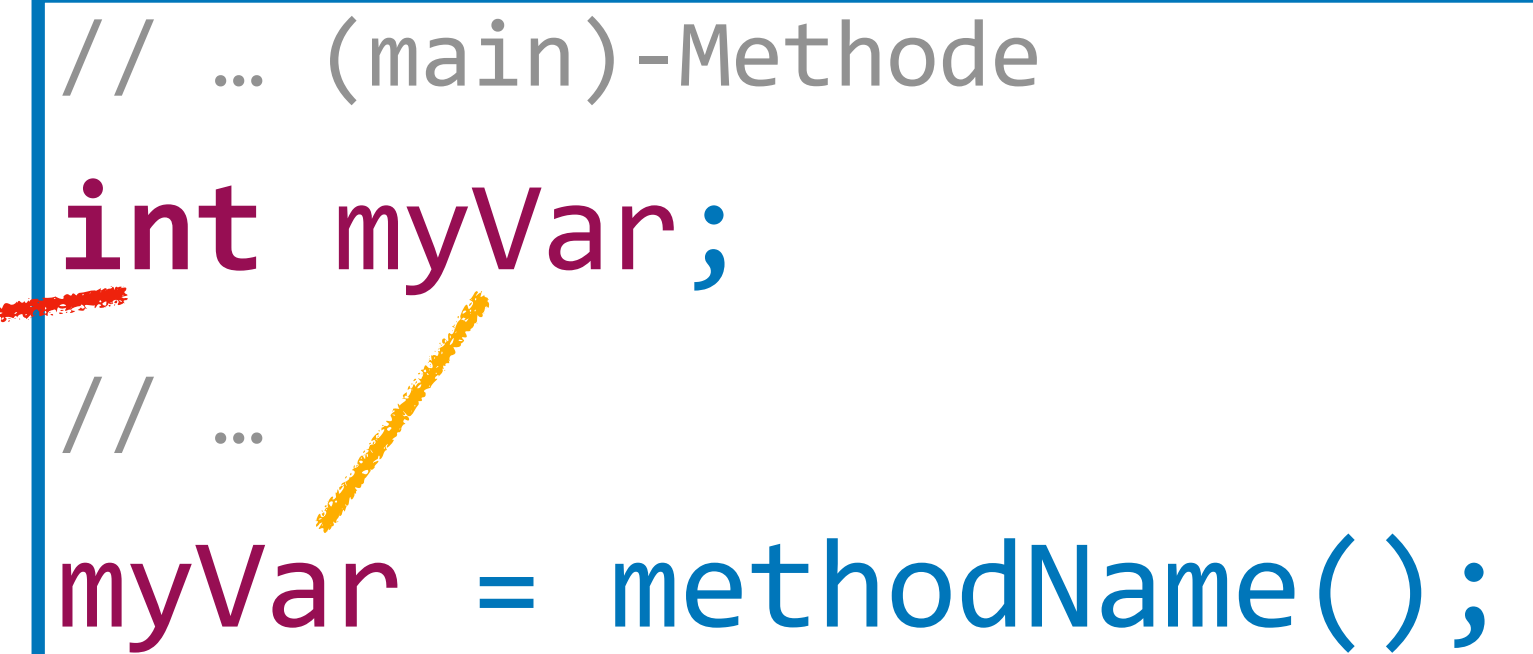
```
public static int methodName() {  
    int x;  
    // ...  
    return x;  
}
```



A red arrow originates from the `return x;` statement and points to the `int myVar;` declaration in the caller block. A green arrow points from the `int x;` declaration to the `return x;` statement.

Aufruf

```
// ... (main)-Methode  
int myVar;  
// ...  
myVar = methodName();
```



A yellow arrow points from the `methodName()` call to the `myVar` variable in the assignment statement. A red arrow originates from the `return x;` statement in the method block and points to the `int myVar;` declaration in this block.

Methoden

Zusammenfassung

- Deklaration <> Aufruf
- Parameter: Datentyp, Anzahl und Reihenfolge
- Rückgabebetyp (Deklaration, return, Aufruf)
- Methodenaufruf an der richtigen Stelle

Beispiele

Methode

```
public static void methodName(String text) {  
    // ...  
}
```

Aufruf

```
// ... (main)-Methode  
_____ // Variable  
methodName(_____);
```

Beispiele

Methode

```
public static _____ isEmpty(String str) {  
    // ...  
}
```

Aufruf

```
// ... (main)-Methode  
String readText;  
// ...  
if (isEmpty(readText)) {  
    // ...  
}
```

Beispiele

Methode

```
public static _____ min(_____) {  
    // ...  
}
```

Aufruf

```
// ... (main)-Methode  
  
// ...  
float lower = min(a,b);
```

Beispiele

Methode

```
public static _____ printUser(_____) {  
    // ...  
    System.out.println(____);  
}
```

Aufruf

```
// ... (main)-Methode  
String name = "Arthur";  
int age = 42;  
bool isHuman = true;  
// ...  
printUser(_____);
```