

Schreiben Sie für die folgende Funktion die Syntax der lambda-Ausdrücke in Java auf.

$$\begin{aligned} f : \mathbb{N} \times \mathbb{N} &\rightarrow \mathbb{N} \\ (m, n) &\mapsto m + n \end{aligned}$$

Und zwar in einer Kurzversion und in einer ausführlichen Version.

$$f : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$$
$$(m, n) \mapsto m + n$$

1 (m, n) -> m + n

2

3 (int m, int n) -> {return m  
+ n;}

<i>Sprache</i>	<i>Wertetabelle</i>	<i>Schreibweise in Mathematik</i>								
Verdoppeln des Wertes	<table><tr><th>x</th><th>f (x)</th></tr><tr><td>2</td><td>4</td></tr><tr><td>6</td><td>12</td></tr><tr><td>15</td><td>30</td></tr></table>	x	f (x)	2	4	6	12	15	30	$x \mapsto f(x)$  $x \mapsto 2 \cdot x$
x	f (x)									
2	4									
6	12									
15	30									

Mathematik:

$$x \mapsto f(x)$$

$$x \mapsto 2 \cdot x$$

Racket:

```
( lambda (x) ( · 2 x ) )
```

Mathematik:

$$x \mapsto f(x)$$

$$x \mapsto 2 \cdot x$$

Java:

```
( int x ) -> { return 2*x ; }
```

Kurzschreibweise:

$$x \rightarrow 2 * x$$

<i>Sprache</i>	<i>Wertetabelle</i>	<i>Schreibweise in Mathematik</i>															
Summe der Quadrate der einzelnen Werte	<table> <tr> <th>x</th><th>y</th><th>f (x, y)</th></tr> <tr> <td>1</td><td>1</td><td>2</td></tr> <tr> <td>1</td><td>2</td><td>5</td></tr> <tr> <td>1</td><td>3</td><td>10</td></tr> <tr> <td>2</td><td>2</td><td>8</td></tr> </table>	x	y	f (x, y)	1	1	2	1	2	5	1	3	10	2	2	8	$(x, y) \mapsto f(x, y)$  $(x, y) \mapsto x^2 + y^2$
x	y	f (x, y)															
1	1	2															
1	2	5															
1	3	10															
2	2	8															

Mathematik:

$$(x, y) \mapsto f(x, y)$$

$$(x, y) \mapsto x^2 + y^2$$

Racket:

```
(lambda (x y) (+ (· x x) (· y y)))
```

## *Anwenden der Funktion auf Werte*

Mathematik:  $x \mapsto f(x)$

$$x \mapsto 2 \cdot x$$

$$\rightsquigarrow f(5)$$

Racket:

```
( (lambda (x) ( · 2 x) ) 5)
```



## Anwenden der Funktion auf Werte

Mathematik:

$$(x, y) \mapsto f(x, y)$$

$$(x, y) \mapsto x^2 + y^2$$

$$\rightsquigarrow f(2, 3)$$

Racket:

```
((lambda (x y) (+ (· x x) (· y y))) 2 3)
```

## *Verwendung von lambda-Ausdrücken*

Teil eines Methodenrumpfs

```
public static UnaryIntFunc fooDouble()  
{  
    return x -> 2*x;  
}
```

## *Verwendung von lambda-Ausdrücken*

Parameter einer anderen Funktion

```
d.foo( x -> 2*x, 5) ;
```

hier: d ist Referenz auf ein Objekt der Klasse, zu der die Methode `foo` gehört

Die Methode `foo` hat zwei Parameter.  
Zuerst vom Typ `UnaryIntFunc` und  
vom Typ `int`.

Gegeben sei folgende Funktion in Racket:

```
1 ( define ( foobar t1 t2 foo bar )  
2   ( lambda ( x y )  
3     ( bar t1 ( foo x ( cons t2 y ) ) )  
4   )  
5 )
```

Wie Sie sehen, sind foo und bar Funktionen. Gegeben sei zudem der folgende beispielhafte Aufruf von foobar:

```
1 ( ( foobar 2 3  
2   ( lambda ( a b ) ( + a ( first b ) ) )  
3   ( lambda ( c d ) ( * c d ) ) )  
4 )  
5 2 (list 5 4 3)  
6 )
```