

# Teil VI

## Datentypen II

## 12. Quiz

*Quiz:* Ich zeige Ihnen einen Typ der Form  $A \rightarrow B$ , sie definieren eine beliebige Funktion dieses Typs.

(*Regeln:* die Funktion sollte total sein, das heißt, sie ist auf allen Eingabewerten definiert, das heißt, sie terminiert für jede zulässige Eingabe und liefert eine Ausgabe des geforderten Typs. Darüber hinaus wäre es schön, wenn die Eingabe auch benutzt wird. Regeln zu brechen, macht natürlich auch Spaß ...)

## 12. Quiz: Teil 1

Zum Aufwärmen ...

- ▶ Wie wär's mit *Bool* → *Bool*?

## 12. Quiz: Teil 1

Zum Aufwärmen ...

- ▶ Wie wär's mit *Bool*  $\rightarrow$  *Bool*?

*let* answer *b* = not *b*

## 12. Quiz: Teil 1

Zum Aufwärmen ...

- ▶ Wie wär's mit  $Bool \rightarrow Bool$ ?

```
let answer b = not b
```

- ▶ ...oder mit  $Nat \rightarrow Nat$ ?

## 12. Quiz: Teil 1

Zum Aufwärmen ...

- ▶ Wie wär's mit  $Bool \rightarrow Bool$ ?

```
let answer b = not b
```

- ▶ ...oder mit  $Nat \rightarrow Nat$ ?

```
let answer n = 2 * n + 1
```

## 12. Quiz: Teil 1

Zum Aufwärmen ...

- ▶ Wie wär's mit  $Bool \rightarrow Bool$ ?

```
let answer b = not b
```

- ▶ ...oder mit  $Nat \rightarrow Nat$ ?

```
let answer n = 2 * n + 1
```

- ▶ ...oder gar  $'a \rightarrow 'a$ ?

## 12. Quiz: Teil 1

Zum Aufwärmen ...

- ▶ Wie wär's mit  $Bool \rightarrow Bool$ ?

```
let answer b = not b
```

- ▶ ...oder mit  $Nat \rightarrow Nat$ ?

```
let answer n = 2 * n + 1
```

- ▶ ...oder gar  $'a \rightarrow 'a$ ?

```
let answer a = a
```

## 12. Quiz: Teil 2

Zweite Aufwärmrunde ...

- ▶ Wie wär's mit  $Bool * Bool \rightarrow Bool$ ?

## 12. Quiz: Teil 2

Zweite Aufwärmrunde ...

- ▶ Wie wär's mit  $Bool * Bool \rightarrow Bool$ ?

```
let answer (b1, b2) = b1 && b2
```

## 12. Quiz: Teil 2

Zweite Aufwärmrunde ...

- ▶ Wie wär's mit  $Bool * Bool \rightarrow Bool$ ?

```
let answer (b1, b2) = b1 && b2
```

- ▶ ...oder mit  $Nat * Nat \rightarrow Nat$ ?

## 12. Quiz: Teil 2

Zweite Aufwärmrunde ...

- ▶ Wie wär's mit  $Bool * Bool \rightarrow Bool$ ?

```
let answer (b1, b2) = b1 && b2
```

- ▶ ...oder mit  $Nat * Nat \rightarrow Nat$ ?

```
let answer (n1, n2) = 2 * n1 + n2
```

## 12. Quiz: Teil 2

Zweite Aufwärmrunde ...

- ▶ Wie wär's mit  $Bool * Bool \rightarrow Bool$ ?

```
let answer (b1, b2) = b1 && b2
```

- ▶ ...oder mit  $Nat * Nat \rightarrow Nat$ ?

```
let answer (n1, n2) = 2 * n1 + n2
```

- ▶ ...oder gar  $'a * 'a \rightarrow 'a$ ?

## 12. Quiz: Teil 2

Zweite Aufwärmrunde ...

- ▶ Wie wär's mit  $Bool * Bool \rightarrow Bool$ ?

```
let answer (b1, b2) = b1 && b2
```

- ▶ ...oder mit  $Nat * Nat \rightarrow Nat$ ?

```
let answer (n1, n2) = 2 * n1 + n2
```

- ▶ ...oder gar  $'a * 'a \rightarrow 'a$ ?

```
let answer1 (a1, a2) = a1
```

```
let answer2 (a1, a2) = a2
```

## 12. Quiz: Teil 3

Etwas spannender ...

- ▶ Wie wär's mit  $Bool \rightarrow (Bool \rightarrow Bool)$ ?

## 12. Quiz: Teil 3

Etwas spannender ...

- ▶ Wie wär's mit  $Bool \rightarrow (Bool \rightarrow Bool)$ ?

```
let answer1 b1 = fun b2 → b1 && b2
```

```
let answer2 b1 b2 = b1 && b2
```

## 12. Quiz: Teil 3

Etwas spannender ...

- ▶ Wie wär's mit  $Bool \rightarrow (Bool \rightarrow Bool)$ ?

```
let answer1 b1 = fun b2 → b1 && b2
```

```
let answer2 b1 b2 = b1 && b2
```

- ▶ ...oder mit  $Nat \rightarrow (Nat \rightarrow Nat)$ ?

## 12. Quiz: Teil 3

Etwas spannender ...

- ▶ Wie wär's mit  $Bool \rightarrow (Bool \rightarrow Bool)$ ?

```
let answer1 b1 = fun b2 → b1 && b2
```

```
let answer2 b1 b2 = b1 && b2
```

- ▶ ...oder mit  $Nat \rightarrow (Nat \rightarrow Nat)$ ?

```
let answer1 n1 = fun n2 → 2 * n1 + n2
```

```
let answer2 n1 n2 = 2 * n1 + n2
```

## 12. Quiz: Teil 3

Etwas spannender ...

- ▶ Wie wär's mit  $Bool \rightarrow (Bool \rightarrow Bool)$ ?

```
let answer1 b1 = fun b2 → b1 && b2
```

```
let answer2 b1 b2 = b1 && b2
```

- ▶ ...oder mit  $Nat \rightarrow (Nat \rightarrow Nat)$ ?

```
let answer1 n1 = fun n2 → 2 * n1 + n2
```

```
let answer2 n1 n2 = 2 * n1 + n2
```

- ▶ ...oder gar  $'a \rightarrow ('a \rightarrow 'a)$ ?

## 12. Quiz: Teil 3

Etwas spannender ...

- ▶ Wie wär's mit  $Bool \rightarrow (Bool \rightarrow Bool)$ ?

```
let answer1 b1 = fun b2 → b1 && b2
```

```
let answer2 b1 b2 = b1 && b2
```

- ▶ ...oder mit  $Nat \rightarrow (Nat \rightarrow Nat)$ ?

```
let answer1 n1 = fun n2 → 2 * n1 + n2
```

```
let answer2 n1 n2 = 2 * n1 + n2
```

- ▶ ...oder gar  $'a \rightarrow ('a \rightarrow 'a)$ ?

```
let answer1 a1 a2 = a1
```

```
let answer2 a1 a2 = a2
```

## 12. Quiz: Teil 4

Jetzt wird's spannend ...

- ▶ Wie wär's mit  $(\text{Bool} \rightarrow \text{Bool}) \rightarrow \text{Bool}$ ?

## 12. Quiz: Teil 4

Jetzt wird's spannend ...

- ▶ Wie wär's mit  $(Bool \rightarrow Bool) \rightarrow Bool$ ?

*let* answer  $f = f$  false && f true

## 12. Quiz: Teil 4

Jetzt wird's spannend ...

- ▶ Wie wär's mit  $(Bool \rightarrow Bool) \rightarrow Bool$ ?

*let* answer  $f = f\ false \ \&\& \ f\ true$

- ▶ ...oder mit  $(Nat \rightarrow Nat) \rightarrow Nat$ ?

## 12. Quiz: Teil 4

Jetzt wird's spannend ...

- ▶ Wie wär's mit  $(Bool \rightarrow Bool) \rightarrow Bool$ ?

```
let answer f = f false && f true
```

- ▶ ...oder mit  $(Nat \rightarrow Nat) \rightarrow Nat$ ?

```
let answer f = f 0 + 1
```

## 12. Quiz: Teil 4

Jetzt wird's spannend ...

- ▶ Wie wär's mit  $(Bool \rightarrow Bool) \rightarrow Bool$ ?

```
let answer f = f false && f true
```

- ▶ ...oder mit  $(Nat \rightarrow Nat) \rightarrow Nat$ ?

```
let answer f = f 0 + 1
```

- ▶ ...oder gar  $('a \rightarrow 'a) \rightarrow 'a$ ?

## 12. Quiz: Teil 4

Jetzt wird's spannend ...

- ▶ Wie wär's mit  $(Bool \rightarrow Bool) \rightarrow Bool$ ?

```
let answer f = f false && f true
```

- ▶ ...oder mit  $(Nat \rightarrow Nat) \rightarrow Nat$ ?

```
let answer f = f 0 + 1
```

- ▶ ...oder gar  $('a \rightarrow 'a) \rightarrow 'a$ ?

```
let rec answer f = f (answer f)
```

## 12. Quiz: Teil 4

Jetzt wird's spannend ...

- ▶ Wie wär's mit  $(Bool \rightarrow Bool) \rightarrow Bool$ ?

```
let answer f = f false && f true
```

- ▶ ...oder mit  $(Nat \rightarrow Nat) \rightarrow Nat$ ?

```
let answer f = f 0 + 1
```

- ▶ ...oder gar  $('a \rightarrow 'a) \rightarrow 'a$ ?

```
let rec answer f = f (answer f)
```

Hmmm ... was ergibt `answer (fun b → not b)`?

## 12. Quiz: Zugabe

Zum Knobeln ...

- ▶ Wie wär's mit  $(\lambda a \rightarrow \lambda' a) \rightarrow (\lambda a \rightarrow \lambda' a)$ ?

## 12. Quiz: Zugabe

Zum Knobeln ...

- ▶ Wie wär's mit  $(\lambda a \rightarrow 'a) \rightarrow (\lambda a \rightarrow 'a)$ ?

*let* answer = DIY

## 12. Quiz: Zugabe

Zum Knobeln ...

- ▶ Wie wär's mit  $(\lambda a \rightarrow 'a) \rightarrow (\lambda a \rightarrow 'a)$ ?

*let answer = DIY*

- ▶ ...oder mit  $List \langle 'a \rangle \rightarrow List \langle 'a \rangle$ ?

## 12. Quiz: Zugabe

Zum Knobeln ...

- ▶ Wie wär's mit  $(\lambda a \rightarrow 'a) \rightarrow (\lambda a \rightarrow 'a)$ ?

*let answer = DIY*

- ▶ ...oder mit  $List \langle 'a \rangle \rightarrow List \langle 'a \rangle$ ?

*let answer = DIY*

## 12. Quiz: Zugabe

Zum Knobeln ...

- ▶ Wie wär's mit  $(\text{'a} \rightarrow \text{'a}) \rightarrow (\text{'a} \rightarrow \text{'a})$ ?

*let answer = DIY*

- ▶ ...oder mit  $\text{List } \langle \text{'a} \rangle \rightarrow \text{List } \langle \text{'a} \rangle$ ?

*let answer = DIY*

- ▶ ...oder gar  $(\text{'a} \rightarrow \text{'b}) \rightarrow (\text{List } \langle \text{'a} \rangle \rightarrow \text{List } \langle \text{'a} \rangle)$ ?

## 12. Quiz: Zugabe

Zum Knobeln ...

- ▶ Wie wär's mit  $(\text{'a} \rightarrow \text{'a}) \rightarrow (\text{'a} \rightarrow \text{'a})$ ?

*let answer = DIY*

- ▶ ...oder mit  $\text{List } \langle \text{'a} \rangle \rightarrow \text{List } \langle \text{'a} \rangle$ ?

*let answer = DIY*

- ▶ ...oder gar  $(\text{'a} \rightarrow \text{'b}) \rightarrow (\text{List } \langle \text{'a} \rangle \rightarrow \text{List } \langle \text{'a} \rangle)$ ?

*let answer = DIY*