

# Funktionale Programmierung (mit Java)

---

Dozent: Prof. Dr. Michael Eichberg  
Kontakt: michael.eichberg@dhbw.de, Raum 149B  
Version: 1.1

---

# 1. Funktionale Programmierung

---

# Kontrollfragen

1.1. Was sind Funktionen höherer Ordnung?

---

1.2. Was sind Lambda-Ausdrücke?

---

1.3. Was steht bei der funktionalen Programmierung im Vordergrund?

2

---

3

## 2. Funktionale Programmierung in Java

---

# Kontrollfragen

## 2.1. Eigenschaften von *Functional Interfaces*

1

Nennen Sie eine wesentliche Eigenschaft von *Functional Interfaces*. Geben Sie ein Beispiel für ein funktionales Interface in Java an.

## 2.2. Nennen Sie mind. drei Möglichkeiten ein funktionales Interface zu implementieren.

2

2.3. Was ist eine Methoden-Referenz? Worauf kann eine Methoden-Referenz verweisen? Beschreiben Sie das allgemeine Muster.

3

# Lambda-Ausdrücke

## 2.4. Typkompatibilität von Lambda-Ausdrücken

Die Methode `forEach` hat die folgende Signatur:

```
void forEach(Consumer<? super T> action);
```

Weiterhin sei gegeben:

```
List<String> names = Arrays.asList("Alice", "Bob", "Charlie");
```

Welche der folgenden Lambda-Ausdrücke kann ich an `forEach` übergeben?

1

1. `names.forEach(name → System.out.println(name));`
2. `names.forEach(name → name.toUpperCase());`
3. `names.forEach(name → { System.out.println(name); return name.toUpperCase(); });`
4. `names.forEach(name → { System.out.println(name); return; });`

Welche der folgenden Objekte kann ich an `forEach` übergeben?

2

5. `Consumer<String> cs = name → System.out.println(name);`
6. `Consumer<Object> co = name → System.out.println(name);`
7. `Consumer<CharSequence> ccs = name → System.out.println(name);` // Hint `String` extends `CharSequence`

# Streams

2.5. Was ist ein Stream?



---

## 2.6. Welche Typen von Streams gibt es?



## 3. Datenstrukturen

# Optional

3.1. Welchem Zweck dient der Datentyp `Optional`?

# Queue / Warteschlange

## 3.2. Warteschlangen verstehen

1. Benennen Sie die essentiellen Eigenschaften einer Warteschlange.
2. Welche Implementierungsstrategien gibt es für Warteschlangen? Erklären Sie diese kurz.

## 3.3. Erstellen Sie ein UML Klassendiagramm für verkettete Listen