

# Kompetenzraster

| Kompetenzband:                                                                                 | HZ | Grundlagen                                                                                                                                                                                                    | Fortgeschritten                                                                                                                                                                                                                                                        | Erweitert                                                                                                                                                                                              |
|------------------------------------------------------------------------------------------------|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Unterschiede zwischen funktionaler Programmierung und anderen Programmierparadigmen aufzeigen. | 1  | AG1: Ich kann die Eigenschaften von Funktionen beschreiben (z.Bsp. pure function) und den Unterschied zu anderen Programmier-Strukturen erläutern (z.Bsp. zu Prozedur).                                       | AF1: Ich kann das Konzept von <code>*immutable values*</code> erläutern und dazu Beispiele anwenden. Somit kann ich dieses Konzept funktionaler Programmierung im Unterschied zu anderen Programmiersprachen erklären (z.Bsp. im Vergleich zu referenzierten Objekten) | AE1: Ich kann aufzeigen wie Probleme in den verschiedenen Konzepten (OO, prozedural und funktional) gelöst werden und diese miteinander vergleichen.                                                   |
| Anforderungen und Design beschreiben                                                           | 1  | BG1: Ich kann den Unterschied zwischen Anforderungen der imperativen Programmierung (definierte Folge von Handlungsanweisungen) und der deklarativen Programmierung (Beschreibung des Endzustandes) erklären. | BF1: Ich kann den Endzustand als Anforderung im Sinne der deklarativen Programmierung beschreiben. (Das gewünschte Ergebnis wird beschrieben statt die Arbeitsschritte.)                                                                                               | BE1: Ich kann Anforderungen aus der imperativen Programmierung in Anforderungen der deklarativen Programmierung transferieren. („klar definierte Abfolge“ transformieren zu „Endergebnis beschreiben“) |
|                                                                                                |    | BG2: Ich kann Elemente des Functional Design erklären. (zBsp. Immutable data types, model, solution, domain of interest, constructors, composable operators)                                                  | BF2: Ich kann für eine Problemstellung ein Functional-Design entwerfen und dabei die Elemente des Functional Designs anwenden.                                                                                                                                         | BE2: Ich kann ein Design einer imperativen Programmierung in ein Design der deklarativen Programmierung transferieren.                                                                                 |
| Funktionale Programmierung umsetzen                                                            | 2  | C1G: Ich kann ein Algorithmus erklären                                                                                                                                                                        | C1F: Ich kann Algorithmen in funktionale Teilstücke aufteilen                                                                                                                                                                                                          | C1E: Ich kann Funktionen in zusammenhängende Algorithmen implementieren                                                                                                                                |
|                                                                                                |    | C2G: Ich kann Funktionen als Objekte behandeln und diese in Variablen speichern und weitergeben.                                                                                                              | C2F: Ich kann Funktionen als Argumente für andere Funktionen verwenden und dadurch höherwertige Funktionen erstellen.                                                                                                                                                  | C2E: Ich kann Funktionen als Objekte und Argumente verwenden, um komplexe Aufgaben zu lösen und den Code sauberer und effizienter zu gestalten.                                                        |
|                                                                                                |    | C3G: Ich kann einfache Lambda-Ausdrücke schreiben, die eine einzelne Operation durchführen, z.B. das Quadrieren einer Zahl oder das Konvertieren eines Strings in Großbuchstaben.                             | C3F: Ich kann Lambda-Ausdrücke schreiben, die mehrere Argumente verarbeiten können.                                                                                                                                                                                    | C3E: Ich kann Lambda-Ausdrücke verwenden, um den Programmfluss zu steuern, z.B. durch Sortieren von Listen basierend auf benutzerdefinierten Kriterien.                                                |

| Kompetenzband:                              | HZ  | Grundlagen                                                                                               | Fortgeschritten                                                                                                                                    | Erweitert                                                                                                                                                                        |
|---------------------------------------------|-----|----------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                             |     | C4G: Ich kann die Funktionen Map, Filter und Reduce einzeln auf Listen anwenden.                         | C4F: Ich kann Map, Filter und Reduce kombiniert verwenden, um Daten zu verarbeiten und zu manipulieren, die komplexere Transformationen erfordern. | C4E: Ich kann Map, Filter und Reduce verwenden, um komplexe Datenverarbeitungsaufgaben zu lösen, wie z.B. die Aggregation von Daten oder die Transformation von Datenstrukturen. |
| Refactoring und bestehenden Code optimieren | 3,4 | DG1: Ich kann einige Refactoring-Techniken aufzählen, die einen Code lesbarer und verständlicher machen. | DF1: Ich kann mit Refactoring-Techniken einen Code lesbarer und verständlicher machen.                                                             | DE1: Ich kann die Auswirkungen des Refactorings auf das Verhalten des Codes einschätzen und sicherstellen, dass das Refactoring keine unerwünschten Nebeneffekte hat.            |
|                                             |     | DG2: Ich kann allgemeine Massnahmen zur Verbesserung der Leistung von Code aufzählen.                    | DF2: Ich kann vorgegebene Massnahmen zur Verbesserung der Leistung von Code umsetzen.                                                              | DE2: Ich kann effiziente Algorithmen, Techniken oder Datenstrukturen auswählen und einsetzen, um die Leistung von Code zu verbessern.                                            |

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

<https://wiki.bzz.ch/modul/m323/kompetenzraster>Last update: **2024/03/28 14:07**