Modulidentifikation

ICT Berufsbildung
Formation professionnelle
Formazione professionale

Modulnummer	323	
Titel	Funktional Programmieren	
Kompetenz	Algorithmen und Teile von Applikationen deklarativ beschreiben und funktional implementieren.	
Handlungsziele	1	Analysiert und beschreibt Anforderungen, so dass eine Realisierung in funktionaler Programmierung erfolgen kann. [g4.1, g4.4]
	2	Implementiert Algorithmen und Teilprobleme von Applikationen effizient nach dem funktionalen Programmierparadigma gemäss vorgegebenen Anforderungen. [g5.2, g5.5]
	3	Verbessert und optimiert bestehenden imperativ implementierten Code durch Anwendung von funktionaler Programmierung (Refactoring). [g5.4, g5.5]
	4	Überprüft Implementierung auf Korrektheit und Qualität. [g6.3, g6.5, g6.6]
Kompetenzfeld	Application Engineering	
Objekt	Anwendung mit 3-5 funktionalen Einheiten	
Modulversion	1.0	
Erstellt am	12.03.2021	

Handlungsnotwendige Kenntnisse

Handlungsnotwendige Kenntnisse beschreiben Wissen, das die kompetente Ausführung der Handlungen eines Moduls unterstützt. Diese Kenntnisse dienen der Orientierung und sind nicht abschliessend definiert. Die daraus folgende Konkretisierung der Lernziele und das Festlegen des Lernwegs für den Kompetenzerwerb sind Sache der Bildungsanbieter.

Modulnummer		323
Titel		Funktional Programmieren
Kompetenz		Algorithmen und Teile von Applikationen deklarativ beschreiben und funktional implementieren.
Handlungsziele und handlungsnotwendige Kenntnisse		
1	1.1	Kennt den Unterschied zwischen imperativer und deklarativer/funktionaler Programmierung (deklaratives Programmierparadigma).
	1.2	Kennt Vorgehensweisen zur deklarativen Beschreibung von Problemen und Endzuständen.
	1.3	Kennt Vor- und Nachteile funktionaler Programmierung.
	1.4	Kennt die Begriffe aus dem funktionalen Programmieren (z.B. Pure-Function, Immutable Data, Lambda-Expression, Function, Closure, Callback, Functor) und versteht diese, um sie korrekt anzuwenden.
	1.5	Kennt Möglichkeiten zur Erarbeitung eines für die funktionale Programmierung geeigneten Realisierungskonzeptes.
2	2.1	Kann funktionalen Code lesen, verstehen, warten und erweitern.
	2.2	Kennt funktionale Programmierkonzepte (z.B. Filter-Map-Reduce) und passende Entwurfsmuster (z.B. Builder-Pattern)
	2.3	Kennt Wege um funktionalen Code verteilt, parallel und nebenläufig auszuführen.
3	3.1	Kennt Mittel um Programmteile daraufhin zu untersuchen, ob diese durch funktionale Implementierung optimiert oder verbessert werden können.
	3.2	Kennt die funktionalen Elemente einer Programmiersprache und kann so Applikationen mit imperativen und deklarativen Programmierparadigma entwickeln.
4	4.1	Kennt Mittel, um funktional implementierten Code zu testen (z.B. Unit-Tests).
	4.2	Kennt geeignete Coderichtlinien und hält diese konsequent ein.
	4.3	Kennt Best-Practices für funktionale Programmierung und kann diese anwenden.

Modulversion	1.0
Erstellt am	12.03.2021