

Modulidentifikation

Modulnummer	319	
Titel	Applikationen entwerfen und implementieren	
Kompetenz	Die Lernenden kennen die Grundlagen des Programmierens, deren Herkunft, ihrer Voraussetzungen und sind in der Lage im Beruflichen Umfeld Probleme zu verstehen und Lösungen dafür zu entwickeln.	
Handlungsziele	1	Erfasst Problemstellungen, entwickelt strukturiert Lösungsansätze und übersetzt sie für die Stakeholder.
	2	Erstellt eine geeignete visuelle Darstellung für die Programmierung von Anforderungen.
	3	Leitet aufgrund der Vorgaben die erforderlichen Daten (Eingabe, Verarbeitung, Ausgabe und ihre Datentypen) ab.
	4	Implementiert die Applikation mit Hilfe von Kontrollstrukturen und selbst erstellten Funktionen.
	5	Hält vorgegebene Konventionen ein, kommentiert den Code und achtet dabei auf die Wartbarkeit.
	6	Interpretiert Mängel (Fehler) in der Software und korrigiert diese.
Kompetenzfeld	Application Engineering	
Objekt	Problemstellungen, welche mit einer Programmiersprache umgesetzt werden können.	
Modulversion	1.0	
Erstellt am	15.03.2021	

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	319	
Titel	Applikationen entwerfen und implementieren	
Kompetenz	Die Lernenden kennen die Grundlagen des Programmierens, deren Herkunft, ihrer Voraussetzungen und sind in der Lage im Beruflichen Umfeld Probleme zu verstehen und Lösungen dafür zu entwickeln.	
Handlungsziele und handlungsnotwendige Kenntnisse		
1	1.1	Kennt den Unterschied und den Zweck von Problemraum zu Lösungsraum.
	1.2	Kennt Instrumente, um Anforderungen Lösungsfrei zu identifizieren.
	1.3	Kennt die Gründe, um den Problemraum und Lösungsraum voneinander zu trennen (z.B Innovationsgewinn, Stressreduktion).
2	2.1	Kennt eine Notation, um den Ablauf einer Programmlogik abzubilden (z. B PAP, Struktogramm, Activity).
	2.2	Kennt den Zweck von Modellen und Diagrammen als Besprechungs- und Umsetzungsgrundlage.
3	3.1	Kennt die Unterscheidung der Datentypen (z. B primitive, komplexe).
	3.2	Kennt die relevanten Datentypen (z. B integer, boolean, array mit einer Dimension).
	3.3	Kennt die Funktionsweise einer Software nach dem EVA Prinzip.
4	4.1	Kennt den grundlegenden Aufbau einer Applikation resp. Programmes.
	4.2	Kennt die Verwendung von Kontrollstrukturen wie Iterationen (vor-/nachgeprüft), Selektion.
	4.3	Kennt Sinn und Zweck von Variablen und Konstanten sowie deren Unterschied.
	4.4	Kennt den Gültigkeitsbereich von Anweisungen und deren Zweck.
	4.5	Kennt die Möglichkeiten von Editoren sowie einer Entwicklungsumgebung, sowie den Umgang mit Fehler- und Warnmeldungen.
	4.6	Kennt den Aufbau von Funktionen (Name, Parameter, Datentypen, Rückgabety) und wie diese in einem Programm korrekt eingesetzt und aufgerufen werden.
	4.7	Kennt den Mehrwert der Auslagerung von Teilen von Code in Funktionen.
5	5.1	Kennt die Bedeutung von Konventionen (z.B. Firmenkonventionen, Sprachkonventionen).
	5.2	Kennt die Bedeutung von Kommentaren und deren Beitrag zu besser wartbarer Software.
6	6.1	Kennt den Zweck, die Funktionen und die Bedienung eines Debugger.
	6.2	Kennt Sinn und Zweck des Testens, sowie die Absicht von einzelnen Testfällen.
	6.3	Kennt ein Verfahren für den statischen Test (z.B. Codereview) von Programmcode und kann die Bedeutung für die Qualität der Software aufzeigen.

Handlungsnotwendige Kenntnisse

Modulversion	1.0
Erstellt am	15.03.2021