



## Modulidentifikation

<b>Modulnummer</b>	<b>404</b>
<b>Titel</b>	Objektbasiert programmieren nach Vorgabe
<hr/>	
Kompetenz	Vorgabe interpretieren, objektbasiert mit einer Programmiersprache implementieren, dokumentieren und testen.
<hr/>	
Handlungsziele	<ol style="list-style-type: none"><li>1 Aufgrund einer Vorgabe den Ablauf darstellen.</li><li>2 Eine Benutzerschnittstelle entwerfen und implementieren.</li><li>3 Erforderliche Daten bestimmen und Datentypen festlegen.</li><li>4 Programmvorgabe unter Nutzung vorhandener Komponenten mit deren Eigenschaften und Methoden, sowie Operatoren und Kontrollstrukturen implementieren.</li><li>5 Beim Programmieren vorgegebene Standards und Richtlinien einhalten, das Programm inline dokumentieren und dabei auf Wartbarkeit und Nachvollziehbarkeit achten.</li><li>6 Programm auf Einhaltung der Funktionalität testen, Fehler erkennen und beheben.</li></ol>
<hr/>	
Kompetenzfeld	Application Engineering
Objekt	Einfaches, objektbasiertes Programm.Zum Beispiel: Taschenrechner
Niveau	1
Voraussetzungen	Keine
Anzahl Lektionen	40
Anerkennung	Eidg. Fähigkeitszeugnis
<hr/>	
Modulversion	1.00



## Handlungsnotwendige Kenntnisse

<b>Modulnummer</b>	<b>404</b>
<b>Titel</b>	Objektbasiert programmieren nach Vorgabe

Kompetenz	Vorgabe interpretieren, objektbasiert mit einer Programmiersprache implementieren, dokumentieren und testen.
-----------	--

### Handlungsnotwendige Kenntnisse

- 1.1 Kennt Darstellungsmöglichkeiten für Programmabläufe Verarbeitungen (z.B. Zustandsdiagramm) und weiss, welche Sachverhalte sich damit abbilden lassen.
- 2.1 Kennt die wichtigsten Regeln für den Aufbau einer Benutzerschnittstelle.
- 2.2 Kennt Lösungsansätze für die Verarbeitung und Validierung von Benutzerinteraktionen.
- 3.1 Kennt den Einsatz und die korrekte Anwendung der wichtigsten Datentypen Ganzzahl, Gleitkommazahl, Zeichen, Zeichenketten und Boolean.
- 4.1 Kennt den grundsätzlichen Aufbau einer Komponente (Eigenschaften, Methoden und Ereignisse) und wie sich diese für die Verarbeitung einsetzen lassen.
- 4.2 Kennt den grundlegenden Aufbau eines Programms und den Einfluss auf den Arbeitsspeicher.
- 4.3 Kennt verfügbare Komponenten einer Programmiersprache mit ihren Eigenschaften, Methoden und Ereignissen sowie Operatoren, und Kontrollstrukturen.
- 5.1 Kennt die wichtigsten Regeln für einen verständlichen Quellcode (sprechende Komponenten- und Variablenbezeichnungen, optische Strukturierung, sinnvolle Kommentare).
- 6.1 Kennt die Elemente eines Testfalls (zu testende Funktion oder Verarbeitung, Eingabedaten, erwartete Ergebnisse)
- 6.2 Kennt die Bedeutung des Testens für die Sicherung der Programmqualität.

Kompetenzfeld	Application Engineering
Objekt	Einfaches, objektbasiertes Programm.Zum Beispiel: Taschenrechner
Niveau	1
Voraussetzungen	Keine
Anzahl Lektionen	40
Anerkennung	Eidg. Fähigkeitszeugnis



**ICT Berufsbildung**  
**Formation professionnelle**  
**Formazione professionale**

Modulversion 1.00



## Leistungsbeurteilungsvorgabe

<b>Modulnummer</b>	<b>404</b>
<b>Modultitel</b>	<b>Objektbasiert programmieren nach Vorgabe</b>
<b>Titel</b>	<b>LBV Modul 404-1 - 2 Elemente - Schriftliche Einzelprüfung / Schriftlicher Test, Praktische Umsetzungsarbeit</b>

---

<b>Übersicht</b>	Im ersten Teil werden die Grundkenntnisse der notwendigen Lerninhalte geprüft. Die Lernenden erkennen ob sie die Anforderungen erfüllen um in Teil 2 bestehen zu können. Im Teil 2 werden alle Handlungsziele sowie persönliche Kompetenzen in einem realistischen Umfeld erlebt. Es sollen nicht nur quantitative (aufbauende) sondern auch qualitätssteigernde Schritte individuell möglich sein.
------------------	---

---

Anzahl Elemente	2
Elementnummer	1

---

<b>Beschreibung</b>	Inhalt: Nutzung von Komponenten der Programmiersprache, sinnvolle Konstruktion und Nutzung von Objekten (Eigenschaften, Methoden, Ereignisse), korrekte Anwendung von Datentypen, Einhalten von Vorgaben mittels UML Ablauf: Praxisorientierte schriftliche Aufgaben mit K1 - K3 Fragen
---------------------	--

---

### Zu überprüfende Handlungsziele

- 1 Aufgrund einer Vorgabe den Ablauf darstellen.
  - 2 Eine Benutzerschnittstelle entwerfen und implementieren.
  - 3 Erforderliche Daten bestimmen und Datentypen festlegen.
  - 4 Programmvorgabe unter Nutzung vorhandener Komponenten mit deren Eigenschaften und Methoden, sowie Operatoren und Kontrollstrukturen implementieren.
- 

<b>Prüfungsform</b>	<b>Schriftliche Einzelprüfung / Schriftlicher Test</b>
---------------------	--

---

Gewichtung in %	50
Richtzeit (Empfehlung)	2

---



Bewertungskriterien	20% HZ1 (Aus Beschreibung einen Ablauf erstellen oder umgekehrt)20% HZ2 (Verwenden von Steuerelementen, Ereignisse)30% HZ3 (Kenntnisse über Datentypen, Bestimmen korrekter Datentypen, Konvertierungen)30% HZ4 (Umsetzung einfacher Klassen nach Vorgabe und deren Anwendung)
Hilfsmittel	Eigenhandschriftliche Notizen im Umfang von zwei A4-Seiten (einseitig bedruckt)IT Handbuch
Praxisbezug	Das Abbilden von realen Gegebenheiten in Klassen als Architekturmerkmal guter Software soll geübt werden.Abläufe korrekt planen und umsetzen mittels Kontrollstrukturen.Sparsamer Umgang mit Speicherplatz und widerspruchsfreie Datenhaltung dank minutiöser Wahl der Datentypen.Basiskonntnisse beim planen und umsetzen von Benutzerschnittstellen.
Anzahl Elemente	2
Elementnummer	2
Beschreibung	Es wird ein Projekt klaren Vorgaben zum Funktionsumfang, Testverfahren und Dokumentation vorgegeben. Anwendung der erarbeiteten Kenntnisse aus Teil 1 sowie zusätzlichen Kompetenzen aus den HZ.Die Erarbeitung des Projekts findet im Unterrichtsgefäss statt.Die Bewertung erfolgt in Form von einer kurzen Produktdemonstration und einem Fachgespräch.HZ5 und HZ6 werden stärker gewichtet, da diese in LB1 nicht vorkommen.
Zu überprüfende Handlungsziele	
<ol style="list-style-type: none"><li>1 Aufgrund einer Vorgabe den Ablauf darstellen.</li><li>2 Eine Benutzerschnittstelle entwerfen und implementieren.</li><li>3 Erforderliche Daten bestimmen und Datentypen festlegen.</li><li>4 Programmvorgabe unter Nutzung vorhandener Komponenten mit deren Eigenschaften und Methoden, sowie Operatoren und Kontrollstrukturen implementieren.</li><li>5 Beim Programmieren vorgegebene Standards und Richtlinien einhalten, das Programm inline dokumentieren und dabei auf Wartbarkeit und Nachvollziehbarkeit achten.</li><li>6 Programm auf Einhaltung der Funktionalität testen, Fehler erkennen und beheben.</li></ol>	
Prüfungsform	Praktische Umsetzungsarbeit



Gewichtung in %	50
Richtzeit (Empfehlung)	8

---

Bewertungskriterien	Es werden nur lauffähige (compilierbare, ausführbare) Programme bewertet.5% Einhaltung zeitliche Vorgabe10% Selbstkompetenz (Selbständiges Arbeiten, Wissensbeschaffung)15% HZ1 Ablauf darstellen (Dokumentation mittels Diagrammen)10% HZ2 Benutzerschnittstelle entwerfen und implementieren10% HZ3 Datentypen bestimmen und festlegen10% HZ4 Programmvorgabe implementieren (Diese ist durch die Aufgabenstellung vorgegeben)20% HZ5 Standards und Richtlinien einhalten, Inline dokumentieren (Clean Code Regeln, Inline Dokumentation Klassen und Methoden sowie Schlüsselpassagen)20% HZ6 Testen, Fehler erkennen und beheben (nur lauffähige Programme werden bewertet, selbständige Qualitätskontrolle)
---------------------	---

---

Hilfsmittel	Programmiersprache und Entwicklungsumgebung sind die gleichen wie in den Modulunterlagen.
-------------	---

---

Praxisbezug	Alle Schritte werden berücksichtigt: Planung, Umsetzung, Dokumentation, Testen (Zeiteinteilung für alle Schritte planen und erleben)Erkennen welche Architekturmöglichkeiten sich bieten und entscheiden welche die geeignetste ist.Umsetzung von Programmlogik und Benutzerschnittstelle sowie deren Zusammenspiel.Verantwortung für das Endprodukt tragen. (testen, verbessern und entscheiden wann die Qualitätsanforderungen erfüllt sind)
-------------	--

---