

Lehrplan

Programmiertechnik

Höhere Berufsfachschule für Automatisierungstechnik

Ministerium für Bildung

Hohenzollernstraße 60, 66117 Saarbrücken
Postfach 10 24 52, 66024 Saarbrücken

Saarbrücken 2010

Hinweis:

Der Lehrplan ist online verfügbar unter
www.saarland.de/lehrplaene.htm

Einleitende Hinweise

Dem vorliegenden Lehrplan liegt die Verordnung – Schul- und Prüfungsordnung – über die Ausbildung und Prüfung an Höheren Berufsfachschulen für Automatisierungstechnik im Saarland (APO-HBFS-AT) vom 7. Juli 2010 zu Grunde.

Der Lehrplan für das Fach Programmierertechnik umfasst in der Unterstufe die Lerngebiete Allgemeine Grundlagen der Programmierung, Elemente der Benutzeroberfläche und der Betriebssystemfunktionen, Kontrollstrukturen und Datenstrukturen. In der Oberstufe werden neben einer Einführung in die Methoden der Softwareentwicklung weiterführende Programmieretechniken behandelt, wie die Paradigmen der objektorientierten Programmierung, dynamische Datenstrukturen, spezielle Elemente der grafischen Benutzeroberfläche, die Anbindung von Datenbanken

Ziel dieses Faches ist es, Absolventen der Höheren Berufsfachschule für Automatisierungstechnik in die syntaktischen und semantischen Regeln höherer, problemorientierter und visuell steuerbarer Software-Entwicklungsumgebungen einzuführen und Methoden und Techniken der Software-Entwicklung zu vermitteln.

Die weitgehend modularen Konzepte moderner Programmentwicklungsumgebungen und die vorhandenen Kontrollstrukturen zwingen den Anfänger, zunächst ein Problem logisch zu durchdringen und es in einem gegliederten Algorithmus aufzuarbeiten. Dieser Schritt ist unbedingt notwendig, um dem Schüler einen Einblick in die Programmierung von grafischen Oberflächen (GUI) zu geben. Der Umgang mit Klassen und Objekten kann gerade bei GUI-basierten Entwicklungsumgebungen gut praktiziert werden.

Moderne objektorientierte Programmierumgebungen sind die bevorzugten Entwicklungssprachen für die Mehrheit der professionellen Programmierer. Aus diesem Grunde sind für den zukünftigen Systeminformatiker (HBFS) fundierte Kenntnisse in Bereich der objektorientierten Sprachen eine Schlüsselqualifikation. Diese Qualifikation verbreitert seinen Einsatzbereich und damit seine Chancen auf dem Arbeitsmarkt.

Insbesondere hat im technischen Bereich die objektorientierte Windows - Programmierung einen hohen Stellenwert. In der Industrie verwendete Sprachen wie Java oder C++ für das .Net Framework bieten quasi patformunabhängige Laufzeitumgebungen mit einer umfangreichen Sammlung von Klassenbibliotheken. Der Lehrplan orientiert sich an diesen Sprachen, ist aber nicht auf sie festgelegt.

Die zeitliche Abfolge der einzelnen Lerngebiete ist als Empfehlung zu verstehen, da die Lerngebiete sowohl aufeinander aufbauen, als auch miteinander querverbunden sind.

Auf nachstehende formale Vorgaben wird verwiesen:

- In seinem Aufbau folgt der Lehrplan einer freien Lernzieltaxonomie, wobei die Lernziele durch Verben beschrieben werden.
- Die Lernziele sind als Groblernziele formuliert.
- Die Zeitrichtwerte sind als vorgeschlagene zeitliche Empfehlung zu verstehen. Sie sind stets als Jahreswochenstunden ausgewiesen, um Vergleiche mit

Schulformen anderer Bundesländer zu ermöglichen.

- Die Lehrplankommission hat Stundenanteile für Wiederholungen und Leistungsüberprüfungen in einem zeitlichen Umfang von ungefähr einem Drittel der ausgewiesenen Zeitrichtwerte vorgesehen.

Saarbrücken, Juli 2010

Lerngebietsübersicht

Lfd. Nr.	Lerngebiet	Zeitrictwert * Stunden
	Unterstufe	
1	Allgemeine Grundlagen der Programmierung	14
2	Elemente der Benutzeroberfläche und der Betriebssystem- funktionen	22
3	Kontrollstrukturen	52
4	Datenstrukturen I	60
5	Einführung in objektorientierte Programmierung	12
Summe		160
	Oberstufe	
6	Softwareentwicklung	20
7	Datenstrukturen II	24
8	Vertiefung der Objektorientierten Programmierung	24
9	Erstellung komplexer grafischer Benutzeroberflächen (GUI)	24
10	Datenbankzugriff	12
11	Projektarbeiten	56
Summe		160

*: Zeitrictwert i. S. eines Vorschlags

Lerngebiet 1: Allgemeine Grundlagen der Programmierung		Zeitrichtwert*: 14 Stunden
Lernziele	Lerninhalte	Hinweise zum Unterricht
1.1 Bedeutung und Aufgabe von Programmiersprachen kennen	<ul style="list-style-type: none"> - Anwendungsbereiche unterschiedlicher Programmiersprachen - Grundsätzliche Unterschiede bei verschiedenen Programmiersprachen 	
1.2 Methoden zur Erstellung von Programmen beherrschen	<ul style="list-style-type: none"> - Problemformulierung, Problemanalyse - Konzeption notwendiger Datenstrukturen - Entwerfen von Algorithmen - Problemlösungsstrategien 	
1.3 Zahlensysteme und deren Bedeutung kennen	<ul style="list-style-type: none"> - Fehlersuche mittels Debugger - dezimales, binäres, hexadezimales, oktales Zahlensystem - Zahlensystemumwandlungen 	

*: Zeitrichtwert i. S. eines Vorschlags

Lerngebiet 2: Elemente der Benutzeroberfläche und der Betriebssystemfunktionen		Zeitrichtwert*: 22 Stunden
Lernziele	Lerninhalte	Hinweise zum Unterricht
2.1 Die Elemente einer Entwicklungsumgebung konfigurieren und benutzen können	<ul style="list-style-type: none"> - Grundlagen der Einbindung der objektorientierten Benutzeroberfläche eines GUI-Systems - Dateihandling, Editorbedienung - Codeerzeugung und –optimierung - Fehlersuche mit Hilfe des Debuggers - Elemente der Windows-Benutzeroberfläche 	Benutzung einer aktuellen und geeigneten Programmierumgebung (Gilt für alle folgenden Lernziele)
2.2 Die Benutzung von Betriebssystemfunktionen kennen	<ul style="list-style-type: none"> - Komponentenbibliotheken - Grafik- und Netzinterface - Bestandteile eines Projektes - Ereignisgesteuerte Programmierung 	

*: Zeitrichtwert i. S. eines Vorschlags

Lerngebiet 3: Kontrollstrukturen

Zeitrichtwert*: 52 Stunden

Lernziele	Lerninhalte	Hinweise zum Unterricht
3.1 Die grundlegenden Sprachelemente und ihre Eigenschaften kennen	<ul style="list-style-type: none">- Kommentare- Bezeichner- Symbole- Reservierte Wörter- Konstanten- Variablen- Zuweisungsoperatoren- Ausdrücke- Arithmetische und logische Operatoren- weitere spezielle Operatoren- Operatorenhierarchie	
3.2 Die Struktur von Programmen kennen	<ul style="list-style-type: none">- Programmkopf- Deklarationsteil- Subroutinen- Hauptprogrammblock	
3.3 Die skalaren Standarddatentypen benutzen und ihre Anwendungsbereiche nennen können	<ul style="list-style-type: none">- Logikdatentypen- Zeichendatentypen- Zeichenkettendatentypen- Ganzzahldatentypen- Gleitkommatentypen- Regeln für die Schreibweise von Bezeichnern- Wertebereiche- Aufzählungstypen- Teilbereichstypen	
3.4 Vertraut sein mit der Typkonvertierung	<ul style="list-style-type: none">- Implizite und explizite Konvertierung von Datentypen	
3.5 Kontrollstrukturen formulieren und benutzen können	<ul style="list-style-type: none">- Bedingte Anweisungen- if-Anweisung- switch-Anweisung- Schleifenanweisungen- while-Anweisung- do-while-Anweisung- for-Anweisung- Sprunganweisungen- continue-Anweisung- break-Anweisung- return-Anweisung	

Lerngebiet 3: Kontrollstrukturen		Zeitrichtwert*: 52 Stunden
Lernziele	Lerninhalte	Hinweise zum Unterricht
3.6 Aufbau und Verwendung von Funktionen kennen und sie benutzen können	<ul style="list-style-type: none"> - Sinn und Zweck von Funktionen - Schnittstelle von Funktionen - Deklaration einer Funktion - Funktionsaufruf - Präprozessordirektiven - Der Datentyp „void“ - Die „return“ – Anweisung 	
3.7 Mit Zeichenketten arbeiten können	<ul style="list-style-type: none"> - Beschreibung einer Zeichenkette - Zeichenkettenoperationen - Zeichenkettenfunktionen - Verwandtschaft zwischen Zeichenketten und Zeichen 	
	-	

*: Zeitrichtwert i. S. eines Vorschlags

Lerngebiet 4: Datenstrukturen		Zeitrichtwert*: 60 Stunden
Lernziele	Lerninhalte	Hinweise zum Unterricht
4.1 Strukturierte Datentypen formulieren und benutzen können	<ul style="list-style-type: none"> - Felddeklaration (Array) - Zugriff auf Feldelemente - Verarbeitung von Feldern - Mehrdimensionale Felder - Strukturdeklaration - Zugriff auf Strukturkomponenten 	
4.2 Mit Zeichenketten arbeiten können	<ul style="list-style-type: none"> - Beschreibung einer Zeichenkette - Zeichenkettenoperationen - Zeichenkettenfunktionen - Verwandtschaft zwischen Zeichenketten und Zeichen 	
4.3 Die Anwendung von Referenzen kennen	<ul style="list-style-type: none"> - Adressoperator - Verweisoperator - Deklaration von Zeigervariablen - Referenzvariablen 	
4.4 Die Gültigkeitsbereiche von Variablen beherrschen	<ul style="list-style-type: none"> - Lokale und globale Variablen - Namensbereiche - Standardnamensbereich 	
4.5 Dateien lesen und schreiben können	<ul style="list-style-type: none"> - Unterschiedliche Dateiformate, Textdatei, strukturierte Datei, - Datei-Zugriffsfunktionen 	
4.6 Die Analyse von Dateien beherrschen	<ul style="list-style-type: none"> - Text- und strukturierte Dateien aufbauen und analysieren - Benutzung von Stringfunktionen 	

*: Zeitrichtwert i. S. eines Vorschlags

Lerngebiet 5: Einführung in objektorientierte Programmierung		Zeitrichtwert*: 12 Stunden
Lernziele	Lerninhalte	Hinweise zum Unterricht
5.1 Allgemeine Grundlagen der objektorientierten Programmierung kennen	<ul style="list-style-type: none"> - Begriff der Klasse - Instanz einer Klasse - Kapselung einer Klasse - Eigenschaften - Methoden - Zugriffsbeschränkungen auf die Elemente einer Klasse - Konstruktor und Destruktor 	
5.2 Einsatzbereiche von globalen und lokalen Daten kennen	<ul style="list-style-type: none"> - Speicherklassen - Zugriffsspezifizierung private, public 	
5.3 Dynamisches Erzeugen von Klasseninstanzen kennen	<ul style="list-style-type: none"> - Operatoren zum Erzeugen und löschen von Instanzen 	

*: Zeitrichtwert i. S. eines Vorschlags

Lerngebiet 6: Softwareentwicklung		Zeitrichtwert*: 20 Stunden
Lernziele	Lerninhalte	Hinweise zum Unterricht
6.1 Modelle der Softwareentwicklung kennen 6.2 Methoden und Werkzeuge der objektorientierten Softwareentwicklung kennen und anwenden können	<ul style="list-style-type: none"> - Wasserfall - Prototyping - V-Modell - OOA, OOD - Lastenheft Pflichtenheft... - UML: - Wichtigste Diagrammtypen - Klassendiagramm, Objektdiagramm, Anwendungsfalldiagramm, Sequenzdiagramm - Aggregation, Komposition, Kardinalität 	

*: Zeitrichtwert i. S. eines Vorschlags

Lerngebiet 7: Datenstrukturen II		Zeitrichtwert*: 24 Stunden
Lernziele	Lerninhalte	Hinweise zum Unterricht
7.1 Referenzen anwenden können	<ul style="list-style-type: none"> - Zeiger: Deklaration, Zuweisung, Zeigerarithmetik, call by reference, Zeiger auf Funktionen, Zeiger auf Objekte - Referenzvariablen: - Deklaration, Zuweisung, Anwendung 	Abhängig von der verwendeten Sprache
7.2 Dynamische Datenstrukturen	<ul style="list-style-type: none"> - Container/Collections: - Listen, verkettete Listen, ArrayList, Vektoren - Stacks, Queues, Sets, Maps 	<p>anwenden können</p> <p>kennen</p>

*: Zeitrichtwert i. S. eines Vorschlags

Lerngebiet 8: Vertiefung der Objektorientierten Programmierung		Zeitrichtwert*: 24 Stunden
Lernziele	Lerninhalte	Hinweise zum Unterricht
8.1 Technik der Vererbung kennen und anwenden können	<ul style="list-style-type: none"> - Syntax, Zugriffsmodifizierer, Basisklasse, abgeleitete Klasse, Aufruf des Konstruktors der Basisklasse, - Mehrfachvererbung 	
8.2 Polymorphie kennen und anwenden können	<ul style="list-style-type: none"> - Überladen von Funktionen/Methoden 	
8.3 Die Bedeutung und Anwendung von Schnittstellen kennen	<ul style="list-style-type: none"> - Schnittstellen definieren und implementieren können - Schnittstelleneinsatz 	
8.4 Abstrakte Klassen anwenden können	<ul style="list-style-type: none"> - Definition, Bedeutung 	
8.5 Schnittstellen verwenden können	<ul style="list-style-type: none"> - Syntax, Bedeutung 	
8.6 Templates kennen und anwenden können	<ul style="list-style-type: none"> - Syntax, Bedeutung, Funktionstemplates, Klassentemplates 	
8.7 Technik der Fehlerbehandlung mit Exceptions kennen	<ul style="list-style-type: none"> - Möglichkeiten, Exceptions auslösen, abfangen, weiterleiten 	

*: Zeitrichtwert i. S. eines Vorschlags

Lerngebiet 9: Erstellung komplexer grafischer Benutzeroberflächen (GUI) Zeitrichtwert*: 24 Stunden		
Lernziele	Lerninhalte	Hinweise zum Unterricht
9.1 Komplexe Steuerelemente einer objektorientierten Sprache kennen und anwenden können	<ul style="list-style-type: none"> - Panels - Split Container - TabControl - ListView - TreeView - Register 	
9.2 Die Verwendung von Menüs und Leisten kennen	<ul style="list-style-type: none"> - Ereignisbehandlung - Aufbau - Menüelemente konfigurieren - Symbolleisten - Kontextmenüs 	
9.3 Einblick in Grafikanwendungen bekommen	<ul style="list-style-type: none"> - Funktionen zeichnen - Bilder anzeigen - Zeichenmethoden 	
9.4 Dialoganwendungen erstellen und anwenden können	<ul style="list-style-type: none"> - Erzeugen, Anzeigen, Konfigurieren von Dialogen - Standarddialoge 	

*: Zeitrichtwert i. S. eines Vorschlags

Lerngebiet 10: Datenbankzugriff		Zeitrichtwert*: 12 Stunden
Lernziele	Lerninhalte	Hinweise zum Unterricht
10.1 Anbindung und Nutzung bestehender Datenbanksysteme implementieren können	- Datenbankklassen und Methoden	

*: Zeitrichtwert i. S. eines Vorschlags

Lerngebiet 11: Projektarbeiten		Zeitrichtwert*: 56 Stunden
Lernziele	Lerninhalte	Hinweise zum Unterricht
11.1 Komplexe, berufsbezogene Aufgabe mit Mitteln der Programmiertechnik selbstständig realisieren können	<p>Komplexe Aufgabenstellung, z.B.</p> <ul style="list-style-type: none"> - Entwicklung eines speziellen Software-Produktes - Entwicklung von fachübergreifenden Anwendungssystemen in komplexen Netzwerkumgebungen oder im Bereich der Mess-, Regel oder der Automatisierungstechnik - Aufbereitung, Einführung und Verwendung von neuen Software-Produkten und Erweiterungen (bspw. .Net Framework in C++) - Recherche - Dokumentation 	

*: Zeitrichtwert i. S. eines Vorschlags