

Lehrplan

Programmierung

Höhere Berufsfachschule für Wirtschaftsinformatik

Unterstufe

Ministerium für Bildung und Kultur

Trierer Straße 33
66111 Saarbrücken

Saarbrücken, Juli 2019

Hinweis:
Der Lehrplan ist online verfügbar unter
www.bildungserver.saarland.de

Einleitende Hinweise

Dem Lehrplan für die Unterstufe im Fach Programmierung liegt die folgende Verordnung zugrunde:

Schul- und Prüfungsordnung - über die Ausbildung und Prüfung an Höheren Berufsfachschulen für Wirtschaftsinformatik im Saarland (APO-HBFS-WI) vom 16. September 1985, zuletzt geändert durch die Verordnung vom 4. Juli 2003

Die didaktische Leitidee hinter dem Lehrplan ist der kompetenzorientierte Unterricht. Dabei wird dem Lehrplan der folgende Kompetenzbegriff zugrunde gelegt:

Kompetenzen werden durch den nachhaltigen Aufbau von Fachwissen, Fertigkeiten und Haltungen vermittelt.

Bei der unterrichtlichen Umsetzung werden konkrete Lernsituationen vorgegeben, in denen die Schülerinnen und Schüler Kompetenzen erwerben und beweisen.

Für das Fach Informatik an beruflichen Schulen gibt es zurzeit keine KMK Bildungsstandards. Daher orientiert sich der Lehrplan an den "Bildungsstandards Informatik für die Sekundarstufe II" der Gesellschaft für Informatik i. d. F. vom 29.1.2016. Diese Standards berücksichtigen die Beschlüsse der KMK vom 1.6.1979 i.d.F. vom 10.5.2007 für "Einheitliche Prüfungsanforderungen im Abitur" im Fach "Berufliche Informatik", ergänzt um die im Lehrplan der Unterstufe erläuterten Operatoren "konfigurieren" und "verwalten".

Die Schülerinnen und Schüler erarbeiten in der Unterstufe im Fach Programmierung allgemeine Datenstrukturierungskonzepte und darauf ablaufende effiziente Algorithmen anhand von klassischen Beispielen der Wirtschaftsinformatik. Die Algorithmen werden unter Verwendung einer objektorientierten Programmiersprache vertiefend eingeübt.

Die Schülerinnen und Schüler lernen im Fach Programmierung Algorithmen zur Lösung von ausgewählten Problemstellungen kennen. Diese theoretischen Kenntnisse werden durch die Erstellung von Programmen in geeigneten Programmierumgebungen praktisch eingeübt und vertieft. Zur Entwicklung der Problemlösungs- und Kommunikationskompetenz sind Phasen der Projektarbeit in Kleingruppen einzuplanen.

Dabei wird insbesondere die Abstraktionsfähigkeit der Schülerinnen und Schüler weiter entwickelt, unter anderem durch Herausarbeitung der für die Lernsituationen relevanten Sachverhalte unter Vernachlässigung unwichtiger Aspekte. Gleichzeitig wird aufgrund der beim Modellieren notwendigen Entscheidungsprozesse das Urteilsvermögen der Schülerinnen und Schüler geschärft, da eine Lernsituation von verschiedenen Standpunkten aus unterschiedlich bewertet werden kann. Am Ende jeder Lernsituation erfolgt eine systematische Überprüfung und kritische Beurteilung der Ergebnisse, sowie die Beurteilung des verwendeten Lösungsweges, was die Fähigkeit zu konstruktiver Kritik fördert.

Der angestrebte Abschluss in der HBFS-WI erfordert in hohem Maße die Fähigkeit des selbstständigen Lernens. Neben dem Fachwissen setzt dies Fähigkeiten voraus, die häufig als Schlüsselqualifikationen bezeichnet werden. Sie umfassen zunächst persönliche Arbeitshaltungen (Anstrengungsbereitschaft, Ausdauer, Gewissenhaftigkeit, Genauigkeit u. a.) und metakognitive Kompetenzen (das Lernen betreffende Regeln und Techniken, Heuristiken, vor allem aber auf tieferes Verständnis zielende Strategien wie z. B. das Variieren von Problemen und Lösungswegen). Zu ihrer Erzielung soll der Unterricht im Fach Systemintegration durch regelmäßige selbständige Erarbeitung fachspezifischer Inhalte einen wesentlichen Beitrag leisten.

Gleiches gilt für den Aufbau sozialer Kompetenzen (Bereitschaft und Fähigkeit zu Kommunikation und Kooperation, zur Übernahme von Verantwortung, u. a.), wenn Perspektivenwechsel, das Verbalisieren eigener Gedanken und Handlungen, das Nachvollziehen der Gedanken anderer und der Austausch rationaler Argumente alternativer Lösungswege geübt werden.

Die Unterrichtsgrundsätze Fächerkooperation, Rechnereinsatz und Projektarbeit haben einen besonderen Stellenwert bei der Lehrplanumsetzung. Beim Entwurf von Lernsituationen soll ein Bezug zu den Fächern Programmierung, Software Entwicklung, Software Anwendungen sowie den wirtschaftswissenschaftlichen Fächern hergestellt werden. Im Rechnerpraktikum bearbeiten die Schülerinnen und Schüler fächerübergreifend Projekte.

Die im Lehrplan angegebenen verbindlichen Lerninhalte stehen grundsätzlich in Zusammenhang mit den Kompetenzerwartungen. Die verbindlichen Inhalte stellen den Mindestkatalog dar, der in dem vorgegebenen Zeitrahmen vermittelt werden muss. Es steht der Lehrkraft offen, weitergehende Inhalte zur Erreichung der angestrebten Kompetenzen zu behandeln.

Des Weiteren enthält der Lehrplan Hinweise für eine mögliche unterrichtliche Umsetzung. Diese stellen Hilfen bei der didaktischen und methodischen Gestaltung des Unterrichts bereit und umreißen die Intensität der Beschäftigung mit den Inhalten.

In den im Lehrplan ausgewiesenen Stundenanteilen sind die Zeiten für Wiederholungen, Leistungsüberprüfungen, Unterrichtsausfall usw. bereits enthalten, sodass der reine Lernstoff zeitlich nur im Umfang von ca. 2/3 angesetzt ist.

Hinsichtlich der Abfolge der einzelnen Lerngebiete ist es empfehlenswert, die beiden ersten Lerngebiete über das gesamte Schuljahr zu unterrichten. Parallel dazu sollte der Unterricht ab Lerngebiet drei einsetzen. Diese Empfehlung berücksichtigt den Sachverhalt, dass die in den Lerngebieten behandelten Inhalte sowohl aufeinander aufbauen als auch untereinander vernetzt sind. Die Lerngebiete bzw. Lerninhalte können ebenso in einer anderen, sachlogischen Reihenfolge vermittelt werden.

Saarbrücken, Juli 2019

Übersicht über die Lerngebiete

Lfd. Nr.	Lerngebiete	Zeitrichtwert in Stunden*
1	Theoretischer Modelle	30
2	Algorithmen	50
3	Grundlagen Programmentwicklung	26
4	Einfachere Programmeentwicklung	40
5	Komplexere Programmentwicklung	54
6	Programme mit grafischer Benutzeroberfläche	120
Summe		320

* Zeitrichtwert i. S. eines Vorschlags

Lerngebiet 1

Theoretische Modelle

Zeitrichtwert: 30 Unterrichtsstunden

Kompetenzen

Die Schülerinnen und Schüler beschreiben die Funktionsweise eines Computers in theoretischen Modellen.

Fachkompetenz

Die Schülerinnen und Schüler

- definieren einen Algorithmus
- erstellen Algorithmen in verschiedenen Darstellungsformen
- überprüfen Algorithmen auf verschiedene Eigenschaften
- beschreiben theoretische Computermodelle.

Verbindliche Lerninhalte

- Algorithmus
- Kontrollstrukturen
- Pseudo-Code
- Grafische Darstellung von Algorithmen
- Rechnermodelle und Rechnerarchitekturen
- Arrays
- Entscheidbarkeit
- Aussagenlogik

Hinweise zum Unterricht bzw. zur Umsetzung

- Die Vor- und Nachteile verschiedener Darstellungsformen sind zu verdeutlichen.
- Die Bedeutung "Strukturierter Programmierung" für die Vermeidung von Fehlern in der Algorithmen Entwicklung ist zu verdeutlichen.
- Rechnermodelle, Entscheidbarkeit und Aussagenlogik sind nur einführend zu behandeln.

Lerngebiet 2

Algorithmen

Zeitrictwert: 50 Unterrichtsstunden

Kompetenzen

Die Schülerinnen und Schüler lösen Problemstellungen mithilfe von Algorithmen und analysieren diese auf bestimmte Qualitätsmerkmale.

Fachkompetenz

Die Schülerinnen und Schüler

- entwickeln Algorithmen für verschiedene Problemstellungen
- ordnen Algorithmen anhand verschiedener Klassifikationen ein
- analysieren Algorithmen im Hinblick auf Korrektheit, Termination und Effizienz.

Verbindliche Lerninhalte

- Sortierverfahren
- Teile-und-herrsche-Verfahren
- Suchalgorithmen
- NP-Vollständige Probleme
- Brute Force Algorithmen
- Greedy Algorithmen
- Determinismus
- Effizienzbetrachtungen
- Asymptotische Notation (O-Notation)
- Worst Case, Best Case und Average Case Laufzeitverhalten
- Verschlüsselungsverfahren
- Rekursion

Hinweise zum Unterricht bzw. zur Umsetzung

- Die Bedeutung des Teile-und-herrsche-Verfahrens für die nebenläufige Programmierung ist zu verdeutlichen.
- Die Größenordnung des Unterschieds zwischen ineffizienten und effizienten Algorithmen hinsichtlich des Ressourcenaufwandes ist zu verdeutlichen.
- Die hohe Komplexität bestimmter Algorithmen als notwendige Voraussetzung für die Sicherheit von Verschlüsselungsverfahren ist zu verdeutlichen.

Lerngebiet 3

Grundlagen Programmentwicklung

Zeitrichtwert: 26 Unterrichtsstunden

Kompetenzen

Die Schülerinnen und Schüler entwickeln einfache Programme.

Fachkompetenz

Die Schülerinnen und Schüler

- erstellen Programme mithilfe einer Entwicklungsumgebung
- wenden Standardklassen und ihre Methoden an
- wenden einfache Basistypen und ihre Operatoren an
- identifizieren und korrigieren Fehler.

Verbindliche Lerninhalte

- Entwicklungsumgebung mit Projektmanagement
- Entwicklerwerkzeuge
- Variablen und Literale
- Basistypen, Wertebereich, Genauigkeit, Typkonvertierungen, Sichtbarkeit
- Operatoren, Prioritäten, Ausführungsreihenfolge, Ausdrücke
- Standardbibliothek

Hinweise zum Unterricht bzw. zur Umsetzung

- Die Lerninhalte werden mit einer Vielzahl von Programmbeispielen praktisch eingeübt.
- Der technische Hintergrund des Kompilierens ist zu verdeutlichen.
- Es ist auf Fehlerquellen, Fehlervermeidung und Fehlerkorrekturen hinzuweisen.

Lerngebiet 4

Einfache Programmentwicklung

Zeitrichtwert: 40 Unterrichtsstunden

Kompetenzen

Die Schülerinnen und Schüler verwenden elementare Programmstrukturen und Funktionen.

Fachkompetenz

Die Schülerinnen und Schüler

- entwickeln kleinere Programme mit Kontrollstrukturen
- implementieren eigene Funktionen
- entwerfen Programme unter Berücksichtigung von Verständlichkeit, Wartbarkeit und Erweiterbarkeit.

Verbindliche Lerninhalte

- Funktionsdeklaration und -aufrufe
- Parameter, Argumente, call by value, call by reference, Rückgabewerte
- Speicherverwaltung
- Testen von Funktionen

Hinweise zum Unterricht bzw. zur Umsetzung

- Die Lerninhalte werden mit einer Vielzahl von Programmbeispielen praktisch eingeübt.
- Es ist auf Fehlerquellen und deren Vermeidung hinzuweisen.
- Es sind Bezüge zu mehreren Programmiersprachen herzustellen.

Lerngebiet 5

Komplexere Programmentwicklung

Zeitrichtwert: 54 Unterrichtsstunden

Kompetenzen

Die Schülerinnen und Schüler entwickeln komplexe Programme.

Fachkompetenz

Die Schülerinnen und Schüler

- implementieren Programme mit komplexen Typen
- implementieren Programme mit spezialisierten Funktionen
- modellieren eine einfache Klassenstruktur
- wenden Methoden und Attribute ihrer Klassen an.

Verbindliche Lerninhalte

- Arrays, Strukturen, Enumerationen
- Funktionen mit benannten Argumenten, Default-Parametern, variablen Parametern
- Rekursion
- Klassen, Attribute, Methoden, Konstruktoren, Instanziierung

Hinweise zum Unterricht

- Die Notwendigkeit zur Strukturierung größerer Programme ist zu verdeutlichen.
- Bei der Klassen-Modellierung sind Bezüge zum Fach Softwaretechnik herzustellen.
- Die prozedurale Programmierung ist der objektorientierten Programmierung gegenüberzustellen.

Lerngebiet 6

Programme mit grafischer Benutzeroberfläche

Zeitrichtwert: 120 Unterrichtsstunden

Kompetenzen

Die Schülerinnen und Schüler entwickeln Programme mit grafischen Benutzeroberflächen unter Verwendung gängiger Bibliotheken.

Fachkompetenz

Die Schülerinnen und Schüler

- erläutern die zur Verfügung stehenden grafischen Steuerelemente
- wenden einfache und komplexe Layouts an
- implementieren dynamische Layouts
- entwickeln Anwendungen über mehrere Applikationsfenster
- erkennen die Notwendigkeit zur Trennung von Daten, Darstellung und Logik.

Verbindliche Lerninhalte

- Planung von Benutzeroberflächen
- Attribute der grafischen Steuerelemente und Layouts
- Ereignisse und Message Loop
- Entwurfsmuster
- Design-Vorlagen für grafische Steuerelemente
- Drag and Drop
- Software Ergonomie

Hinweise zum Unterricht

- Die Entwurfsmuster sollen an einfachen Beispielen verdeutlicht werden.
- Bei der Planung von Benutzeroberflächen sind heterogene Systeme zu berücksichtigen.
- Auf plattformübergreifende GUI Bibliotheken und Internationalisierung ist hinzuweisen.