

Lehrplan

Programmierung

Höhere Berufsfachschule
für Wirtschaftsinformatik
Oberstufe
SP Anwendungsentwicklung

Ministerium für Bildung und Kultur
Trierer Straße 33
66111 Saarbrücken

Saarbrücken, Juli 2020

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

Einleitende Hinweise

Dem Lehrplan für die Oberstufe 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. Januar 2016. Diese Standards berücksichtigen die Beschlüsse der KMK vom 1. Juni 1979 i.d.F. vom 10. Mai 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.

In der Oberstufe werden die aus der Unterstufe bereits bekannten Algorithmen im Bereich der Programmiermethodik insbesondere um Datenstrukturen ergänzt. Im Bereich der Programmierung werden die Kenntnisse weiter vertieft und um das Thema Datenbankanbindung ergänzt. 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 HBFSWI 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 Lerninhalte stehen grundsätzlich in Zusammenhang mit den Kompetenzerwartungen. Die Inhalte sind verbindlich, sie 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.

Die Lerngebiete 1 und 2, 3 bis 5 sowie 6 bis 8 bilden jeweils eigenständige Einheiten, die unabhängig voneinander unterrichtet werden können. Dies sollte bei der Unterrichtsplanung berücksichtigt werden.

Saarbrücken, Juli 2020

Übersicht über die Lerngebiete

Lfd. Nr.	Lerngebiet	Zeitrichtwert (UStd.*)
1	Komplexe Datenstrukturen und Algorithmen	60
2	Formale sprachen und Automaten	20
3	Fortgeschrittene Programmierkonstrukte	40
4	Fortgeschrittene Anwendungsfälle	40
5	Entwurf und Analyse komplexer Softwarestrukturen	40
6	Grundlagen Datenbankanbindung	20
7	Benutzeroberflächen mit Datenbankanbindung	50
8	Web-Anwendungen mit Datenbankanbindung	50
Gesamtstunden		320

* Zeitrichtwert i. S. eines Vorschlags

Lerngebiet 1: Komplexe Datenstrukturen und Algorithmen

Zeitrictwert: 60 Unterrichtsstunden

Die Schülerinnen und Schüler wenden Datenstrukturen zum effizienten Verwalten von Datenbeständen an.

Die Schülerinnen und Schüler

- erläutern gängige Datenstrukturen
- wenden Standardoperationen auf den Datenstrukturen an
- implementieren Algorithmen auf Datenstrukturen
- wenden komplexe Datenstrukturen problemorientiert an.

Lerninhalte

- Stack und Queues
- Zeigerkonzept
- Einfach und doppelt verkettete Listen
- Bäume und Algorithmen
- Binärbaum und balancierter Bäume
- Hashtabelle
- Gerichtete, ungerichtete Graphen und Algorithmen

Hinweise zum Unterricht bzw. zur Umsetzung

- Verkettete Listen sind als grundlegende Datenstruktur für Zeigeralgorithmen zu verwenden.
- Für den Einsatz komplexer Datenstrukturen sind konkrete Anwendungsbeispiele zu behandeln.
- Die Effizienzvorteile verschiedener Baum-Modelle sind herzuleiten.

Lerngebiet 2: Formale Sprachen und Automaten

Zeitrictwert: 20 Unterrichtsstunden

Die Schülerinnen und Schüler kennen Grundlagen formaler Sprachen und deren Anwendung.

Die Schülerinnen und Schüler

- definieren formale Sprachen mithilfe einer Grammatik
- erstellen Automaten für vorgegebene Situationen
- erläutern die Anwendung formaler Sprachen und Automaten in der Informatik.

Lerninhalte

- Grammatiken
- Deterministische endliche Automaten
- Algorithmen auf Zeichenketten

Hinweise zum Unterricht bzw. zur Umsetzung

- Automatentheorie und formale Sprachen sind nur einführend zu behandeln.
- Die Verbindung zwischen theoretischer und praktischer Informatik soll ansatzweise dargestellt werden.
- Die Bedeutung effizienter Algorithmen auf Zeichenketten ist hervorzuheben.

Lerngebiet 3: Fortgeschrittene Programmierkonstrukte

Zeitrictwert: 40 Unterrichtsstunden

Die Schülerinnen und Schüler implementieren Programme unter Anwendung fortgeschrittener Programmierkonstrukte.

Die Schülerinnen und Schüler

- wenden objektorientierte Programmier Techniken an
- wenden komplexe Konstrukte zur Manipulation von Programmfluss und Daten an
- erläutern Grundtechniken der parallelen Programmierung
- identifizieren und korrigieren Fehler.

Lerninhalte

- Vererbung
- Interfaces
- Behandlung von Laufzeitfehlern
- Sichtbarkeit und Zugriffsrechte
- Speichermanagement
- Threading
- Asynchrone Programmierung

Hinweise zum Unterricht bzw. zur Umsetzung

- Die Vorteile der Programmierung gegen Interfaces sind hervorzuheben.
- Es sind verschiedene Konzepte der Laufzeitfehlerbehandlung zu diskutieren.

Lerngebiet 4: Fortgeschrittene Anwendungsfälle

Zeitrictwert: 40 Unterrichtsstunden

Die Schülerinnen und Schüler implementieren Programme für häufig auftretende Anwendungsfälle.

Die Schülerinnen und Schüler

- verwalten Daten unter Berücksichtigung von Persistenz und Kompatibilität
- wenden Datenstrukturen und Algorithmen zur effizienten Programmierung an
- überprüfen das Laufzeitverhalten von Programmen.

Lerninhalte

- Lesen und Schreiben von Dateien
- Dateiformate
- Zeichenkodierung
- Manipulation von Zeichenketten
- Nutzung komplexer Datenstrukturen
- Verschlüsselung
- Testen
- Messen von Systemauslastung

Hinweise zum Unterricht bzw. zur Umsetzung

- Bei der Modellierung sind Bezüge zu Planungstechniken der Softwaretechnik herzustellen.
- Die Bedeutung der Analyse von bestehendem Programmcode ist hervorzuheben und einzuüben.

Lerngebiet 5: Entwurf und Analyse komplexer Softwarestrukturen

Zeitrichtwert: 40 Unterrichtsstunden

Die Schülerinnen und Schüler analysieren und erstellen komplexe Softwareentwürfe.

Die Schülerinnen und Schüler

- modellieren komplexe Programme
- nutzen externe Programmierschnittstellen
- erweitern bestehende Programme.

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: Grundlagen Datenbankanbindung

Zeitrictwert: 20 Unterrichtsstunden

Die Schülerinnen und Schüler erstellen einfache Konsolenprogramme mit Datenbankanbindung.

Die Schülerinnen und Schüler

- planen Programme mit Datenbankanbindung unter funktionalen und sicherheitstechnischen Anforderungen
- implementieren Programme mit Datenbankanbindung
- testen Programme mit Datenbankanbindung.

Lerninhalte

- Architektur von Datenbankanwendungen
- Zugriff auf Datenbanken
- Abfragen und Anzeigen von Daten
- Ändern von Daten
- Hinzufügen und Löschen von Datensätzen
- Parametrisierte Anweisungen

Hinweise zum Unterricht

- Nutzen und Notwendigkeit verschiedener Datenbankarchitekturen sind zu besprechen.
- Der Bezug zum Fach Softwaretechnik in der Unterstufe ist herzustellen.

Lerngebiet 7: Grafische Benutzeroberflächen mit Datenbankanbindung

Zeitrichtwert: 50 Unterrichtsstunden

Die Schülerinnen und Schüler implementieren grafische Benutzeroberflächen mit Datenbankanbindung.

Die Schülerinnen und Schüler

- planen GUI-Anwendungen mit Datenbankanbindung
- implementieren GUI-Anwendungen mit Datenbankanbindung
- testen GUI-Anwendungen mit Datenbankanbindung.

Lerninhalte

- Verbinden von Datenbankinhalten mit GUI-Elementen
- Anzeigen und Ändern von Daten
- Hinzufügen und Löschen von Datensätzen
- Navigieren in Datenbeständen
- Suche in Datenbeständen

Hinweise zum Unterricht

- Die Beziehungen im Datenmodell sind beim Softwareentwurf zu berücksichtigen.
- Der Bezug zum Fach Softwaretechnik in der Unterstufe ist herzustellen.

Lerngebiet 8: Web-Anwendungen mit Datenbankanbindung

Zeitrictwert: 50 Unterrichtsstunden

Die Schülerinnen und Schüler implementieren Web-Anwendungen mit Datenbankanbindung.

Die Schülerinnen und Schüler

- planen Web-Anwendungen mit Datenbankanbindung
- implementieren Web-Anwendungen mit Datenbankanbindung
- testen Web-Anwendungen mit Datenbankanbindung.

Lerninhalte

- Architektur von Web-Anwendungen
- Sicheres Login
- Trennung von Programmlogik, Darstellung und Daten
- Web-Frameworks
- Web-APIs, Web-Services
- Internationalisierung
- Benutzerfreundlichkeit
- Barrierefreiheit

Hinweise zum Unterricht

- Die Parallelen zur GUI-Programmierung sind zu verdeutlichen.
- Aktuelle Entwicklungen in der Web-Programmierung sind zu behandeln.
- Bezüglich der Konfiguration von Web-Anwendungen ist auf das Fach Systemintegration hinzuweisen.