

Lehrplan



Informatik



Fachschule für Technik
Fachrichtung Kraftfahrzeugtechnik

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 vorliegenden Lehrplan im Fach Informatik in der Fachschule für Technik, Fachrichtung Kraftfahrzeugtechnik liegt die Verordnung – Schul- und Prüfungsordnung – über die Ausbildung und Prüfung an Fachschulen für Technik (APO-T) vom 11. Juni 2003 i.d.F. vom 8. Juli 2020 zu Grunde.

Als Schulform folgt die Fachschule für Technik der KMK-Rahmenvereinbarung über Fachschulen, Beschluss der Kultusministerkonferenz vom 7. November 2002 i.d.F. vom 22. März 2019.

Durch die Neugestaltung des Bildungsgangs in der Fachschule für Technik, Fachrichtung Kraftfahrzeugtechnik, Schwerpunkt Alternative Antriebe wird der Entwicklung alternativer Antriebskonzepte als zentraler Baustein eines nachhaltigen Mobilitätskonzeptes bei stetig steigender Verkehrsleistung Rechnung getragen. Die Stundentafel in dem Bildungsgang wurde entsprechend angepasst und die Lehrpläne einzelner Fächer wurden überarbeitet bzw. neu gestaltet.

Durch den Unterricht im Fach Informatik werden den Schülerinnen und Schülern grundlegende Elemente einer strukturierten Programmiersprache vermittelt, die sie zur Umsetzung von Lösungsalgorithmen anwenden. Durch die Benutzung einer integrierten Entwicklungsumgebung wird die Arbeitsweise eines Compilers verdeutlicht. Dabei werden die Vorteile und die Mächtigkeit von objektorientierten Programmier- und Datenstrukturen erklärt. Bei der Programmierung von grafischen Benutzerschnittstellen wird dieses Wissen angewandt, vertieft und mit der systematischen Vorgehens- und Testweise bei der Software-Erstellung geschult.

Im Sinne des Erwerbs einer erweiterten beruflichen Handlungskompetenz nimmt die Förderung der Fachkompetenz in der Fachschule für Kraftfahrzeugtechnik einen besonderen Stellenwert ein. Die zu vermittelnden Lerninhalte bauen auf einer einschlägigen berufsschulischen Ausbildung auf. Die rasche technische Entwicklung in Verbindung mit dem schnellen Wandel normativer Vorgaben erfordern von den Schülerinnen und Schülern ein hohes Maß an Flexibilität und die Fähigkeit, eigenverantwortlich zu lernen.

Auf nachfolgende formale Vorgaben wird verwiesen:

- Der Lehrplan ist in drei Lerngebiete unterteilt. Eine generalisierende Beschreibung der Kernkompetenz am Ende des Lernprozesses ist jedem Lerngebiet vorangestellt. Diese wird durch weitere Kompetenzbeschreibungen präzisiert, denen Lerninhalte an separater Stelle zugeordnet sind. Die im Lehrplan formulierten Kompetenzen bieten Freiräume, die eine zeitnahe Einbindung neuer Technologien in den Unterrichtsprozess ermöglichen.
- Die Zeitrichtwerte sind als vorgeschlagene zeitliche Empfehlung zu verstehen. Sie sind stets als Jahresstunden ausgewiesen, um Vergleiche mit den Fachschulen für Technik anderer Bundesländer zu ermöglichen.
- Stundenanteile für Wiederholungen und Leistungsüberprüfungen sind in den ausgewiesenen Gesamtstunden berücksichtigt.

Übersicht über die Lerngebiete

Grundstufe		
Lfd. Nr.	Lerngebiet	Zeitrichtwert (UStd.*)
1	Grundlagen der strukturierten Programmierung	80
Gesamtstunden		80

Fachstufe		
Lfd. Nr.	Lerngebiet	Zeitrichtwert (UStd.*)
2	Objektorientierte Programmiertechniken	40
3	Datenmanagement und Visualisierung	40
Gesamtstunden		80

* Zeitrichtwert i. S. eines Vorschlags

Lerngebiet 1: Grundlagen der strukturierten Programmierung

Zeitrichtwert: 80 Unterrichtsstunden

Die Schülerinnen und Schüler erfassen einfache informationsbezogene Aufgabenstellungen, formulieren dazu Lösungsalgorithmen und realisieren diese in einer Entwicklungsumgebung für eine objektorientierte Programmiersprache.

Die Schülerinnen und Schüler formulieren Problemstellungen und zeigen anhand von Struktogrammen Strategien und Lösungswege auf, die sie mit Algorithmen beschreiben. Sie bedienen die integrierte Entwicklungsumgebung einer objektorientierten Programmiersprache um lauffähige Programme zu erstellen.

Sie beschreiben die Arbeitsweise eines Compilers und leiten daraus die Bedeutung für die Syntax der grundlegenden Sprachelemente ab.

Sie wenden die passenden Datentypen zur Abbildung der Problemstellung aus und verarbeiten diese Daten in passenden Operationen und Ausdrücken in Programmen.

Sie nutzen Kontrollstrukturen, um ihren Programmen mehr Mächtigkeit und Flexibilität zu geben.

Sie lagern Programmteile in Funktionen/Prozeduren aus, sorgen für eine fehlersichere Über- und Rückgabe von Parametern, um ihren Programmen Ordnung und Struktur zu geben.

Sie schreiben einfache Programmteile, die zur Interaktion dienen und die Verarbeitung und Manipulation von Zeichenketten möglich machen.

Lerninhalte

- **Programmerstellung:** Problemformulierung, Problemanalyse, Datenstrukturen, Entwerfen von Algorithmen, Problemlösungsstrategien, Programmablaufpläne, Struktogramme, Integrierte Entwicklungsumgebung
- **Grundlegende Sprachelemente:** Kommentare, Bezeichner, Symbole, reservierte Wörter, Konstanten, Variablen
- **Standard-Datentypen:** Zeichen, Zeichenketten, Ganzzahlen, Gleitkommazahlen, Wahrheitsdatentypen, Felder, Zeiger
- **Ausdrücke:** Operatoren, Zeichenketten-Ausdrücke, arithmetische Ausdrücke, Logische Ausdrücke
- **Kontrollstrukturen:** Bedingte Programmverzweigungen, Zählwiederholungen, Wiederholstrukturen, Mehrwegauswahl
- **Funktionen / Prozeduren:** Zweck von Funktionen und Prozeduren, Funktions- und Prozedurkopf, Arten der Parameterübergabe, Funktions- und Prozeduraufruf, lokale und globale Variablen
- **Zeichenketten:** Beschreibung und Aufbau von Zeichenketten, Zeichenkettenoperatoren, Zeichenkettenfunktionen und Prozeduren, Umwandlungsfunktionen

Hinweise zum Unterricht bzw. zur Umsetzung

Auswahl einer Struktogramm-Form oder -Norm, mit der Aufgabe und Probleme strukturiert werden

Auswahl einer objektorientierten Programmiersprache und Installation der passenden integrierten Entwicklungsumgebung

Übungsaufgaben, die sich zunächst an für die Schülerinnen und Schüler bekannten Problemstellungen orientieren

Lerngebiet 2: Objektorientierte Programmiertechniken

Zeitrictwert: 40 Unterrichtsstunden

Die Schülerinnen und Schüler beschreiben, analysieren und wenden erweiterte Strukturen der objektorientierten Programmiertechnik an.

Die Schülerinnen und Schüler erklären das Konzept der objektorientierten Programmierung am Beispiel der Instanz eines Klassentyps. Sie definieren Klassen, zugehörige Objekte und wenden das Konzept der Vererbung zur Schaffung neuer Klassen an. Sie analysieren bestehende Anwendungen sowie deren Klassen und erweitern deren Funktionalitäten durch geeignete Methoden zur Ereignissteuerung.

Sie entwickeln und realisieren anforderungsgerechte Softwarelösungen mit Hilfe eines visuellen Software-Entwicklungssystems selbstständig.

Sie unterscheiden und benennen die Konzepte zur Kapselung und Vererbung und wenden diese zielorientiert an.

Lerninhalte

- Weiterführende Datentypen: Aufzählungstypen, Teilbereichstypen, Struktur von Feldern, Felddeklarationen, Verarbeitung von Feldern, Mehrdimensionale Felder
- Objektorientierte Problemlösungen und Ereignissteuerung
- Klassen als Datentypen von Objekten: Methoden und Eigenschaften als Klassenkomponenten, Kapselung, Vererbung, Arten von Methoden, Aktivierung und Implementierung von Methoden, Konstruktoren und Destruktoren, Klassenoperatoren
- Instanz eines Klassentyps

Hinweise zum Unterricht bzw. zur Umsetzung

Integrierte Entwicklungsumgebungen bieten die Möglichkeit, Basisanwendungen und Objektbibliotheken für unterschiedliche Programmarten zu nutzen. Diese Objekte sollen mit Ereignissteuerungen erweitert oder verändert werden und so die erworbenen Fähigkeiten zur Gestaltung von benutzerfreundlicheren Programmen angewendet werden. Dabei sind die Vorteile der Kapselung von Eigenschaften und Methoden herauszustellen und anzuwenden.

Lerngebiet 3: Datenmanagement, Visualisierung und Test

Zeitrictwert: 40 Unterrichtsstunden

Die Schülerinnen und Schüler programmieren grafische Benutzerschnittstellen, Funktion zum Lesen und Speichern von Daten und testen ihre Programme systematisch.

Die Schülerinnen und Schüler nutzen die objektorientierten Methoden zur Programmierung grafischer Benutzeroberflächen von einfachen Programmen und wenden das Wissen um Zeichenkettenverarbeitung bei der Eingabe von Daten an.

Sie programmieren Funktionalitäten zum Einlesen von gespeicherten Daten und der Abspeicherung von Programmerngebnissen.

Sie berücksichtigen Aspekte der Qualitätssicherung bei der Entstehung von Software und definieren passende Testszenarien.

Lerninhalte

- Grafische Benutzerschnittstellen: Interaktionsobjekte einsetzen, Ereignissteuerung zur Verbesserung der Bedienbarkeit von Software
- Handhabung von Dateien: Textdatei, Datei-Zugriffsfunktionen bzw. –Prozeduren
- Qualitätssicherung und systematisches Testen von Software-Projekten

Hinweise zum Unterricht bzw. zur Umsetzung

Entsprechend der verwendeten Entwicklungsumgebung sind geeignete Bibliotheken und Templates zu verwenden, um die wesentlichen Interaktionsobjekte einer grafischen Benutzeroberfläche einzusetzen und entsprechende Ereignis-Routinen dafür zu programmieren. Ebenso für den lesenden und schreibenden Zugriff auf externe Dateien.

Unter 'Testen' ist die Gesamtheit der Maßnahmen zur Überprüfung der Softwarequalität (inkl. Planung, Vorbereitung, Steuerung, Durchführung, Dokumentation usw.; siehe auch Definitionen) zu verstehen.