

Gruppe 10: Was sind Lernziele? (Definition)

Die Lernziele geben an, was Lernende nach einem bestimmten Unterrichtsabschnitt können sollen. (Wikipedia)

Lernziele beschreiben Eigenschaften, die der Lernende nach erfolgreicher Lernerfahrung erworben haben soll. Sie erfüllen didaktische Funktionen für die Entwicklung von Lehrmaterialien und haben Einfluss auf die Inhalte und Methodenwahl. Als Bestandteil der Lehrinhalte können sie den Adressaten zusätzliche Informationen geben. (Mager)

Ein Lernziel ist eine sprachliche Formulierung, die beschreibt, welche Lernergebnisse und welches Verhalten innerhalb einer festgelegten Zeitspanne erwartet werden bzw. erreicht werden sollen. (Weicker)

Warum ist die Definition von Lernzielen wichtig?

- Benutzer sieht von Beginn an, was vermittelt werden soll
- Voraussetzung für eine zielgerichtete Lernprogrammaufbereitung
- Überprüfung der Lernergebnisse
- Roter Faden
- Auf die Zielgruppe bezogen
- Bewusste Vorbereitung
- Grundlage für Auswahl der Inhalte
- Reduktion aufs Wesentliche
- Darstellungen und Methoden zur Vermittlung
 - Lernziele ermöglichen eigen Wege zum Ziel für den Lernenden
 - Lernen ist planbar

Beispielhafte Lernziele zum Thema „Lehrerweiterbildung im Fach Informatik“

- Allgemeine Lernziele

Neben der Beherrschung von Rechnersystemen auf Anwenderebene soll die für den Informatik-Unterricht ausgebildete Lehrkraft die Fähigkeit besitzen, in grundlegenden Bereichen der Informatik

- Probleme und Problemlösungen auf verschiedenen Abstraktionsstufen zu formulieren und zu behandeln
- informatische Sachverhalte unter sauberer Verwendung der Fachsprache präzise beschreiben zu können
- gesellschaftliche Aspekte reflektieren zu können

- Lernziele zu Programmierung, Programmiersprachen, Algorithmen

Die Lehrkraft soll

- grundlegende Prinzipien der systematischen Programmierung kennen
- diese Prinzipien in mindestens zwei verschiedenen Programmiersprachen unterschiedlicher Abstraktionsebenen in deren Grundkonstrukte umsetzen können
- in diesen Sprachparadigmen Spezifikationen von Daten- und Programmstrukturen auf jeweils angemessenem Abstraktionsniveau erstellen können
- die unterschiedlichen Sprachebenen bei der Formulierung von Spezifikation und Realisierung sauber abgrenzen können;
- wesentliche Datentypen und Algorithmen entwerfen und analysieren;
- dabei die Grundkonzepte rekursiver Techniken sicher beherrschen und
- mit den verwendeten Programmierumgebungen souverän umgehen können
- eine angemessen breite Palette grundlegender Algorithmen programmiersprachenfrei formulieren und sie zur Lösung von Problemen einsetzen können
- über fundierte Einsichten in die Grundbegriffe der theoretischen Informatik verfügen und deren Bezug zu Konzepten der erlernten Programmiersprachen herstellen können
- den Begriff der (maschinellen) Berechenbarkeit präzisieren können;
- Problemklassen benennen und ihre algorithmische Komplexität herleiten
- auf dieser Basis die prinzipiellen Grenzen der Lösbarkeit bestimmter Probleme anhand geeigneter Beispiele aufzeigen können

Beispielhafte Kernideen „Fit im (Straßen-) Verkehr“

- Navigation anhand von Verkehrszeichen
 - Rechtsabbiegepfeil für „Weiter“
 - Notrufsäule für Hilfe
 - Ausfahrthinweis für Verlassen des Programms
- Animierte Darstellung von Verkehrssituationen
- Bildschirm wird zur Windschutzscheibe (man sitzt quasi im Auto)
 - Man fährt durch das Programm
 - Gas und Bremse für vorwärts und stoppen
 - Ortsschilder zu den Aufgaben
 - Navigation mit Hilfe des Lenkrads und der Blinker (plus Navigationssystem)