

Informatik

Der Informatikunterricht führt in die Gesetze und Prinzipien ein, welche die Welt der Information bestimmen. Er befasst sich mit der Erforschung und der Gestaltung automatisierter Abläufe und zeigt Möglichkeiten und Grenzen der digitalen Informationsverarbeitung auf.

In fast allen Wissenschaftsgebieten und Berufen werden Grundlagenkompetenzen in Informatik vorausgesetzt. Das Fach Informatik leistet damit einen wichtigen Beitrag sowohl zur allgemeinen Studierfähigkeit als auch zur vertieften Gesellschaftsreife.

Ziel des Informatikunterrichts ist es, den SuS grundlegende Kenntnisse und Fähigkeiten zu vermitteln, die es ihnen ermöglichen, ein Verständnis dafür zu entwickeln, wie die digitale Welt "hinter den Kulissen" funktioniert.

Der Unterricht fördert die Entwicklung exakter Planungs- und Problemlösestrategien und abstrahierendes und vernetztes Denken. Durch das unmittelbare Feedback beim Programmieren fördert er kreatives Denken, die Freude am Experimentieren und ermöglicht es Strategien zu Fehlersuche und zur Optimierung zu entdecken.

Richtziele

Grundhaltungen

- Bereitschaft, Informatikmittel einzusetzen und sich für ihre Funktionsweise zu interessieren.
- Sich aufgrund der Bau- und Funktionsweise von Computern Möglichkeiten und Grenzen der Computertechnologie überlegen.
- Bereitschaft, reale Probleme zu abstrahieren, mittels eines Algorithmus' zu modellieren und durch ein Programm zu simulieren.
- Sich überlegen, wie man grosse Datenmengen automatisiert verarbeiten und daraus neue Informationen gewinnen kann.
- Sich für die Vernetzung der heutigen Systeme interessieren und kritische Fragen zur Sicherheit dieser Systeme stellen.
- Immer wieder von Neuem den sich laufend ändernden Einfluss der Informatik auf Wissenschaft, Wirtschaft und Gesellschaft reflektieren

Grundkenntnisse

- Bedeutung der Informatik als Wissenschaft kennen.
- Unterschiedliche Darstellungsformen von Algorithmen kennen.
- Kenntnisse darüber, wie ein Computer Informationen codiert.
- Unterschiedliche Aspekte von Codierungen kennen (Komprimierung, Fehlererkennung, Fehlerkorrektur, Verschlüsselung).
- Kenntnisse grundlegender Konzepte, die in vielen Programmiersprachen vorkommen.
- Kenntnisse verschiedener Möglichkeiten zur Repräsentation und Strukturierung von Daten.
- Grundlegendes Verständnis des Aufbaus von Netzwerken.

Grundfertigkeiten

- Reale Systeme abstrahieren und modellieren können.
- Fähigkeit, grosse Probleme in kleinere Probleme zu zerlegen.
- Einen Algorithmus in einer Programmiersprache implementieren können.
- Über Strategien verfügen, um Fehler bei der Programmierung selbständig zu finden.
- Projektorientiert arbeiten können.

Grobziele

Unterstufe (1./2. Klasse)

In der Unterstufe steht der Einstieg ins Programmieren im Vordergrund. Mittels graphischer Systeme, die den SuS ein unmittelbares Feedback geben, erlernen die SuS die grundlegenden Befehle einer Programmiersprache, die Modularisierung von Problemen und Strategien zur Fehlersuche und Optimierung.

Die SuS können..

- für einfache Problemstellungen algorithmische Lösungswege suchen und auf Korrektheit prüfen sowie verschiedene Lösungswege vergleichen.
- Wiederholungen mithilfe von Schleifen programmieren.
- ein Programm in Unterprogramme modularisieren (Funktionen, Prozeduren, Methoden).
- mithilfe von Parametern Anweisungen verallgemeinern und diese zur Steuerung von Abläufen in einem Programm einsetzen.
- Variablen zur Speicherung von Daten einsetzen.
- verschiedene Datentypen (Zahlen, Zeichen, Wahrheitswerte) unterscheiden und beim Programmieren richtig einsetzen.
- erklären, wie Daten verloren gehen und die wichtigsten Massnahmen nennen, um sich davor zu schützen.
- Ausdrücke formulieren und die (Zwischen-) Resultate mittels Zuweisungsoperator speichern.
- Benutzereingaben anfordern und Ausgaben an die Benutzenden zurückgeben.
- in ihrem Umfeld Verzweigungen erkennen, beschreiben und strukturiert darstellen (z.B. mittels Flussdiagramm) sowie Verzweigungen zur Steuerung von Abläufen in einem Programm einsetzen.
- ein Programm als Abfolge vordefinierter Anweisungen erklären und begründen, warum ein Computer in nicht antizipierten Situationen unter Umständen anders reagiert als erwartet.
- Fehler als syntaktische oder semantische identifizieren und beim Programmieren Strategien zur Fehlererkennung und -behebung formulieren und praktisch umsetzen.

Oberstufe (3./4. Klasse)

Die Programmierkenntnisse aus der Unterstufe werden ausgebaut und mit neuen Programmierkonzepten ergänzt. Es wird ein Verständnis dafür geschaffen, wie Informationen codiert werden und welche Eigenschaften eine Codierung haben kann. Ebenso wichtig ist die Frage, wie Informationen effizient sortiert und durchsucht werden können. Die Funktionsweise und Sicherheit von vernetzten Systemen wird thematisiert. Der Einfluss der Informatik auf Wissenschaft, Wirtschaft und Gesellschaft wird reflektiert.

- Algorithmen und Programmierung
 - Elementaraussagen korrekt formulieren und mit logischen Operatoren verknüpfen.
 - unterschiedliche Arten von Verzweigungen einsetzen und durch die korrekte Formulierung von Bedingungsprüfungen den Programmablauf steuern.
 - Zähler- und bedingungsgesteuerte Schleifen unterscheiden und situationsgerecht einsetzen.
 - Parameter an Unterprogramme übergeben und Resultate ins Hauptprogramm zurückgeben.
 - die Lebensdauer und Sichtbarkeit von Variablen abschätzen und diese in einem Programm je nach Bedarf als lokale oder globale Variable einsetzen.
 - Daten mittels Listen strukturieren, erfassen, suchen und automatisiert auswerten.
 - die Funktionsweise eines Standard-Algorithmus (z.B. Such- und Sortieralgorithmus) erklären sowie diesen an realen Daten anwenden und Optimierungsstrategien formulieren.
 - unter Einsatz von Pseudo-Zufallszahlen ein einfaches Zufallsexperiment durchführen und die Resultate interpretieren.
 - Fehler als syntaktische oder semantische identifizieren und beim Programmieren Strategien zur Fehlererkennung und -behebung formulieren und praktisch umsetzen.

- Daten
 - die Funktions- und Bauweise eines Computers sowie deren Implikationen auf die Datencodierung erklären.
 - das Binär- und das Hexadezimalsystem erklären und kennen die Einheiten Bit und Byte.
 - die Darstellung von Zeichen, Zahlen und Farben im binären/hexadezimalen System erklären.
 - ein Kompressionsverfahren «von Hand» durchführen und situationsadäquat zwischen verlustfreier und verlustbehafteter Kompression entscheiden.
 - Baum- und Netzstrukturen (z.B. Ordnerstruktur auf dem Computer, Stammbaum, Mindmap, Website) erklären und diese zum Suchen von Information verwenden.
 - einfache Datenbanken abfragen (z.B. mit SQL) und das zugehörige Datenbankmodell (z.B. Relationen-Modell) verstehen und beurteilen.
 - an einem konkreten Beispiel aus Rohdaten Information und aus Information Wissen generieren; sie können erklären, welche Entscheidungen sie gefällt und worauf sie sich bezogen haben.
 - die grundsätzliche Funktionsweise einer KI erklären.

- Vernetzung, Sicherheit und Gesellschaft
 - die gesellschaftliche Bedeutung von KIs reflektieren.
 - das Internet als Infrastruktur von seinen Diensten unterscheiden (z.B. WWW, E-Mail, Internettelefonie, Soziale Netzwerke).
 - eine einfache Netzwerkarchitektur (z.B. Teilnetze und die sie verbindenden Komponenten) beschreiben.
 - an Beispielen die Bedeutung von Protokollen zur Adressierung und Übermittlung von Daten beschreiben.
 - die Begriffe «Klartext», «Schlüssel», «Code» definieren, sowie unterschiedliche Verschlüsselungsmethoden beschreiben und «von Hand» durchführen.
 - die wichtigsten Sicherheitsbedrohungen im Internet einschätzen und sich mittels geeigneter Massnahmen davor schützen.