

Kontexte Jahrgangsstufen 9 und 10

Kontext „Mein digitaler CO ₂ -Fußabdruck“	2
Kontext „Mit Strom steuern, rechnen und speichern!“	5
Kontext „Neuron als Baustein einer lernenden KI“	8
Kontext „KI-basierter Bildverarbeitungssensor“	12
Kontext „Die Welt im Modell“	15
Kontext „Spacebug – Algorithmen für das Weltall“	19

Kontext „Mein digitaler CO₂-Fußabdruck“

Zielsetzung dieses Kontextes

In dieser Unterrichtsreihe sollen die Schülerinnen und Schüler ein kritisches Bewusstsein für die Auswirkungen der Digitalisierung auf Umwelt und Klima entwickeln und den eigenen Umgang mit der Ressource Energie reflektieren, indem sie den Energieverbrauch von Internetdiensten untersuchen.

Zunächst erfahren die Schülerinnen und Schüler, was passiert, wenn sie im Internet surfen und welche Auswirkung eine Suche im Internet oder das Nutzen von Streaming-Diensten hat. Der Zusammenhang zwischen dem Nutzen von Internetdiensten, dem Energieverbrauch (exemplarisch in den Rechenzentren) und dem damit verbundenen CO₂-Fußabdruck wird hergestellt.

Anschließend dokumentieren die Schülerinnen und Schüler ihr eigenes Nutzungsverhalten von Online-Diensten über einen definierten Zeitraum, indem sie die Anzahl von Suchanfragen und das Nutzen von Streaming-Diensten erfassen. Die gesammelten Daten werden in der Klasse ausgewertet und kritisch reflektiert.

Abschließend reflektieren die Schülerinnen und Schüler ihr individuelles Online-Verhalten und formulieren konkrete Maßnahmen zur Reduktion des digitalen CO₂-Fußabdrucks und

diskutieren Herausforderungen und Chancen eines nachhaltigen digitalen Verhaltens.

Vorbedingungen für die unterrichtliche Durchführung





Die Schülerinnen und Schüler kennen die Bedeutung des Begriffes CO₂-Fußabdruck und seine Auswirkungen auf das Klima.

Materialien

Zugriff auf

[inf-schule | Informatik und Gesellschaft » Globale Auswirkungen](#)

Mein digitaler CO₂-Fußabdruck (Zeitansatz: 8h)

Kompetenzen	Anregungen und Hinweise
<p> Fragen stellen und Vermutungen über informatische Sachverhalte äußern</p> <p> Auswirkungen bei der Einführung eines Informatiksystems diskutieren</p> <p> Wechselwirkungen zwischen Informatiksystemen und ihrer gesellschaftlichen Einbettung benennen</p> <p> Beschreiben, wie Menschen vor und nach der Einführung von Informatiksystemen leben und arbeiten</p>	<p>Internet: Energiebedarf und CO₂-Fußabdruck</p> <ul style="list-style-type: none"> ○ Fragestellungen z.B.: <ul style="list-style-type: none"> • Wie viel Strom verbrauchen Rechenzentren? • Womit ist der CO₂-Fußabdruck des Internets vergleichbar? • Womit ist der Energiebedarf / CO₂-Fußabdruck von Suchanfragen vergleichbar? • Womit ist der CO₂-Fußabdruck von Streaming vergleichbar? ▶ Video: „Wie die Digitalisierung Strom frisst“ auf inf-schule ○ Welche Webseiten und Apps nutze ich täglich? <ul style="list-style-type: none"> • Dokumentation des Aufrufs von Webseiten und Nutzung von Apps über einen gewissen Zeitraum • Kritische Reflexion des eigenen Verhaltens ○ Der Energiebedarf der Informations- und Kommunikationstechnik (IKT) <ul style="list-style-type: none"> • Recherche zum Energiebedarf der IKT in Deutschland ▶ Gruppenarbeit mit abschließender Präsentation ○ Energieeffizientere Produkte und Rebound-Effekte <ul style="list-style-type: none"> • Welche Energielabel gibt es? • Was sind Rebound-Effekte? • Umweltfreundlich oder Rebound-Effekt? Beurteilung von Situationen (s. inf-schule) <p>Schaden meine Suchanfragen der Umwelt?</p> <ul style="list-style-type: none"> ○ Betrachtung einer konkreten Suchmaschine (z.B. Google, Bing ...) <ul style="list-style-type: none"> • Recherche (z.B. Sitz der Rechenzentren, Energieverbrauch von Rechenzentren) • Dokumentation über einen gewissen Zeitraum: Wie viele Anfragen wurden getätigt und wie hoch war dabei in etwa der CO₂-Fußabdruck? (s. inf-schule) • Kritische Reflexion: welche Suchanfragen hätten vermieden werden können? ○ Alternative Suchmaschinen <ul style="list-style-type: none"> • Welche Suchmaschinen sind weniger umweltschädlich? ▶ Vergleich der Suchmaschinen nach verschiedenen Kriterien

Schadet mein Streamingverhalten der Umwelt?

- Wo findet Streaming statt? (Streaminganbieter, Videoplattformen, Social Media ...)
- Dokumentation über einen gewissen Zeitraum: Wie lange wurden Videos oder Musik gestreamt und wie hoch war dabei in etwa das Datenvolumen und der resultierende CO₂-Fußabdruck? (s. inf-schule)
 - ▶ Podiumsdiskussion mit verschiedenen Rollen (z.B. Anwender, Umweltschützerin, Klimaaktivist, Energieversorgerin (s. inf-schule))
- Kritische Reflexion: Wie kann man Streaming-Dienste energieeffizienter nutzen?

Welche Konsequenzen hat das für mein eigenes Handeln?

- Erstellung einer Übersicht mit Tipps und Hinweisen, wie man selbst nachhaltiger im Internet und mit mobilen Geräten unterwegs sein kann.

Kontext „Mit Strom steuern, rechnen und speichern!“

Zielsetzung dieses Kontextes

In dieser Unterrichtsreihe erhalten die Schülerinnen und Schüler einen Einblick in die technischen Grundlagen der Digitaltechnik, indem sie binäre Eingangssignale mit Logikgattern zu Ausgangssignalen verarbeiten. In einem weiteren Schritt speichern sie diese Signale z.B. mit einem RS-Flipflop.

Der Zugang zum Thema erfolgt über schülernahe und alltagsrelevante Beispiele (z.B. Fahrstuhlsteuerung). Mit Hilfe von Simulationen entwickeln die Lernenden einfache Problemlösungen, indem sie an diesen Beispielen Ein- und Ausgänge identifizieren, mit binären Zuständen codieren und Logikgatter zur Verknüpfung verwenden. Die Analogie zu elektrischen Schaltkreisen kann helfen, eine modellhafte Vorstellung von Logikgattern zu entwickeln.

Im nächsten Schritt erforschen die Lernenden den prinzipiellen Aufbau einfacher Rechenwerke. Ausgehend vom Binärsystem verwenden sie den Halbaddierer zur Addition und erarbeiten dessen Aufbau.

Abschließend wird die Speicherung von Zuständen thematisiert: An schülernahen Beispielen erkennen die Schülerinnen und Schüler die Notwendigkeit, binäre Zustände zu speichern. Um

das Prinzip der Speicherung in der Digitaltechnik zu verstehen, reicht es aus, den Aufbau und die Funktionsweise eines RS-Flipflops zu betrachten.











Materialien

Simulationen wie z.B. LogicSim und DSIMWeb

Zugriff auf

[inf-schule | Funktionsweise eines Rechners / Digitaltechnik](#)

Mit Strom steuern, rechnen und speichern! (Zeitansatz: 12h)

Kompetenzen	Anregungen und Hinweise
<p> arithmetische und logische Operationen kennen und verwenden</p> <p> Operationen auf Daten sachgerecht durchführen</p> <p> die Grundlagen des Aufbaus von Informatiksystemen und deren Funktionsweise verstehen</p> <p> wesentliche Hardwarekomponenten durch ihre Kenngrößen charakterisieren</p> <p> informatische Modelle zu gegebenen Sachverhalten erstellen</p> <p> Informatiksysteme und Anwendungen unter dem Aspekt der zugrunde liegenden Modellierung betrachten</p> <p> Sachverhalte analysieren und angemessene Modelle erarbeiten</p> <p> das Modellverhalten durch zielgerichtete Änderungen beeinflussen</p> <p> Fragen zu einfachen informatischen Sachverhalten formulieren</p> <p> informatisches Wissen nutzen, um Fragen zu komplexeren Problemstellungen zu formulieren</p>	<p>Wie steuert man in der Informatik mit Strom?</p> <ul style="list-style-type: none"> ○ Zugang über mögliche Situationen im Alltag: z.B. „Wann darf der Fahrstuhl losfahren?“ / „Wann läuft die Heckenschere?“ (s. inf-schule.de) ○ Simulation mit Grundgattern (NOT / AND / OR / XOR) <ul style="list-style-type: none"> • sprachliche Beschreibung („Fahrstuhl fährt, wenn ...“) • Identifizieren von Ein- und Ausgängen (Sensoren/Schalter als Eingänge, Aktoren als Ausgänge) • Abbildung der Eingänge und Ausgänge auf lediglich zwei Zustände – Schalter an und aus / Motor an und aus • Codierung der sprachlich beschriebenen Zustände als 0 und 1 • Zusammenfassung aller möglichen Fälle in einer Wahrheitstabelle • sprachliche Beschreibung der Wahrheitstabelle (Aussage „Motor läuft ist wahr – Wert 1, wenn...“), auch mit Hilfe von Variablen („Schalter 1 – Variable e1“) • die sprachlichen Beschreibung (und/oder/nicht) den Grundgattern zuordnen • Simulation von Schaltungen mit Eingaben (Schalter), Verknüpfung (Grundgatter) und Ausgabe (LED) • Formale Darstellung als Schaltterm ○ Klärung des „umgangssprachlichen“ ODER als XOR ○ Analogie zu Stromkreisen bilden und logischen Verknüpfungen der Programmiersprache ○ Üben (auch Digitalo-Karten) ○ Optional: NAND, NOR als weitere Grundgatter, die aus den bekannten Gattern zusammengesetzt werden können. <p>Wie rechnet man mit Strom?</p> <ul style="list-style-type: none"> ○ Wiederholung Binärsystem ○ „Warum ist 1+1=10?“ – Addition im Binärsystem und Addition im Zehnersystem <ul style="list-style-type: none"> • Übertrag und Summe ▶ Summe meint beim Halbaddierer nur die letzte Ziffer



informatische Sachverhalte unter Benutzung von Fachsprache mündlich und schriftlich sachgerecht darstellen



Diagramme, Grafiken und Modelle nutzen, um sich informatische Sachverhalte selbstständig zu erarbeiten

- Umsetzen der Addition in einer Schaltung
 - Festlegen von Eingängen (Ziffern der Binärzahl) und Ausgängen (Übertrag und Summe)
 - Wahrheitstabelle für beide Ausgänge
 - Simulation als Schaltung
- Halbaddierer als logischer Baustein (Modularisierung) und sein innerer Aufbau aus Grundgattern
- Optional: Erweiterung zum Volladdierer und Addierwerk

Wie speichert man mit Strom?

- Zugänge aus dem Alltag: „Alarmanlage“ oder „Der heiße Draht“
 - „Wie merkt man sich, dass der Draht berührt wurde?“
oder: „Anforderung Fußgängerampel, Aufzug“
- Rückkopplung als Phänomen erfahren
 - ▶ Simulation mit Grundgattern:
Ausgang mit Eingang verbinden und Auswirkung beobachten
- Erweiterung durch Reset
- RS-Flipflop (Modularisierung) als Speichereinheit für ein Bit
 - Gängige Speichergrößen
 - Flüchtigkeit des Speichers als Problem erkennen
 - Weitere Flipflop-Arten sind nicht intendiert.

Kontext „Neuron als Baustein einer lernenden KI“

Zielsetzung dieses Kontextes

Neuronale Netze sind eine „Form“ Künstlicher Intelligenz (KI), die vom biologischen Nervensystem inspiriert wurde. Deren Leistung besteht darin, Zuordnungen zwischen Eingaben und Ausgaben herzustellen.

Um KI zu entmystifizieren, werden in diesem Kontext nur einzelne Neuronen als Bausteine eines solchen künstlichen neuronalen Netzes betrachtet. Dabei werden Neuronen als universelle (didaktisch reduzierte) Bausteine behandelt, die Logikgatter simulieren sollen.

Solche künstlichen Neuronen können z.B. die Funktionsweise des UND-Gatters automatisiert „erlernen“, indem ihr Schwellenwert und ihre Kantengewichte verändert werden.

An diesem vereinfachten Beispiel eines einzelnen Neurons wird das Prinzip von überwachtem Lernen verdeutlicht.

Vorbedingungen für die unterrichtliche Durchführung

Aus dem Kontext „Mit Strom steuern, rechnen und speichern“ (Digitaltechnik): Logikgatter und Wahrheitstabelle

Aus dem Kontext „Entscheide wie eine KI!“ (Entscheidungsbaum): Schwellenwert

Materialien

Zugriff auf

https://dev.inf-schule.de/KI_Basis

Fächerübergreifender Unterricht

Bezug zu Lehrplan Naturwissenschaften (7-10): Biologie TF 7
Reiz, Nervenzelle, neuronales Netz

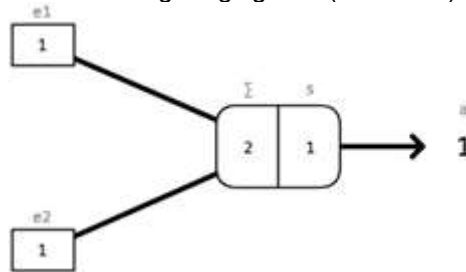
Neuron als Baustein einer lernenden KI (Zeitansatz: 8h)

Kompetenzen	Anregungen und Hinweise
<p>die Grundlagen des Aufbaus von Informatiksystemen und deren Funktionsweise verstehen</p> <p>Operationen auf Daten verstehen und interpretieren</p> <p>arithmetische und logische Operationen verwenden</p> <p>informatische Modelle zu gegebenen Sachverhalten erstellen</p> <p>strukturieren Sachverhalte durch zweckdienliches Zerlegen und Anordnen,</p> <p>erkennen und nutzen Verbindungen innerhalb und außerhalb der Informatik.</p>	<p>Das Neuron in der Informatik</p> <p>○ Was ist ein Neuron in der Informatik?</p> <ul style="list-style-type: none"> • Ein künstliches Neuron ist ein vereinfachtes Modell eines biologischen Neurons. Es bildet Teile der biologischen Funktion nach. • Vereinfacht wird es zunächst als Blackbox betrachtet, die nur zwei Eingänge (deren Werte nur 0 oder 1 sein können) und einen Ausgang hat. • Wenn ein Neuron „feuert“, so ist der Ausgang 1. „Feuert“ es nicht, so ist der Ausgang 0. Neuronen können dabei verschiedene Verhaltensweisen annehmen. • Die wesentlichen Merkmale eines künstlichen Neurons (Schwellenwert, Kantengewichte) werden im Folgenden schrittweise eingeführt. <div style="text-align: center;"> </div>

Wie funktioniert ein künstliches Neuron?

- Erste Annäherung: Ein Neuron soll sich wie ein UND-Gatter verhalten

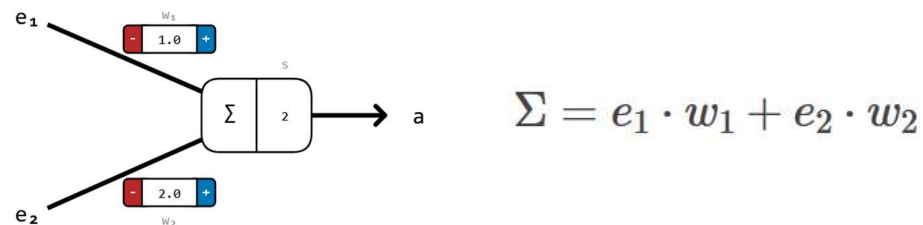
- Wiederholung: Logikgatter (z.B. UND) und deren Wertetabelle



- Nutzung eines (vereinfachten) künstlichen Neurons ohne Kantengewichte und mit Schwellenwert 1
- Übung: Ein künstliches Neuron soll die Funktion eines Oder-Gatters annehmen (Veränderung des Schwellenwerts, Angeben mehrerer geeigneter Schwellenwerte z.B. auch Kommazahlen)
- Notwendigkeit zur Erweiterung des Aufbaus eines Neurons:
z.B. beim Schwellenwert 2 kann das Neuron kein UND- oder ODER-Gatter simulieren

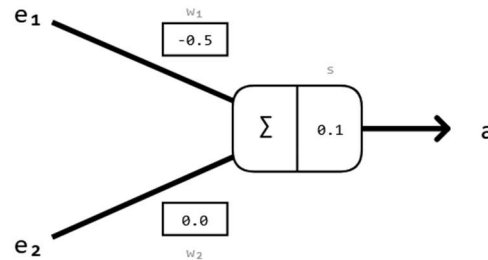
- Nächster Schritt: Einführung der Kantengewichte und der gewichteten Summe

- Ein positives Kantengewicht hat verstärkenden, ein negatives einen hemmenden Einfluss auf die Aktivierung.
- Wie erreichen wir durch Verändern der Kantengewichte, dass das Neuron bei Schwellenwert 2 trotzdem ein UND- bzw. ODER-Gatter nachahmt?
- Definition und Bedeutung der gewichteten Summe im Hinblick auf das Feuern des Neurons
- Weitere Übungsbeispiele zum Verhalten von Neuronen (auch mit negativen Schwellenwerten und Kantengewichten)



Wie lernt ein künstliches Neuron?

- Herausforderung:
 - Das Neuron muss dazu gebracht werden, die Werte für Kantengewichte (und Schwellenwert) selbständig einzustellen, also zu lernen.
 - Im Fehlerfall werden Kantengewichte verändert, sodass der Fehler beim nächsten Mal kleiner wird und irgendwann verschwindet.
- Das selbstlernende Neuron (schrittweise Erarbeitung des Lernalgorithmus' eines Neurons)



e_1	e_2	Σ	a	Vorgabe
0	0	0.0	0	0
0	1	0.0	0	0
1	0	-0.5	0	1
1	1	-0.5	0	1

- Ausgangspunkt ist ein Neuron mit zufälligen Kantengewichten und eine Wertetabelle mit gewünschtem Verhalten.
- Dabei wird eine vereinfachte Lernregel (s. inf-schule) verwendet, die zunächst nur die Kantengewichte verändert.
- Übungen und Experimente mit anderen Startwerten
- Dabei verdeutlichen, dass nicht alle Wahrheitstabellen (z.B. NAND, XOR) realisiert werden können.
- Optional:
 - Auch den Schwellenwert lernen lassen
 - Grenzen: Was geht denn überhaupt? (z.B. XOR)
 - Einsatz von Tabellenkalkulation zur Umsetzung der Lernregeln
 - Ändern der Lernrate
 - Ausblick: Verknüpfen von Neuronen
- Alternative Beispiele außerhalb der Digitaltechnik

Kontext „KI-basierter Bildverarbeitungssensor“

Zielsetzung dieses Kontextes

In dieser projektorientierten Unterrichtsreihe entwickeln die Schülerinnen und Schüler ein kritisches Bewusstsein für KI-basierte Anwendungen in Technik und Gesellschaft, indem sie einen KI-basierten Bildverarbeitungssensor in verschiedenen technischen Problemstellungen einsetzen und die Chancen und Risiken dieser Technologie analysieren und bewerten.

Der Zugang zu diesem Thema erfolgt über Alltagserfahrungen mit Bild- und Gesichtserkennung. Im nächsten Schritt der Unterrichtsreihe wird in die KI-basierte Bildverarbeitungstechnologie am Beispiel der „HuskyLens“ eingeführt. Dabei steht die Vermittlung der grundlegenden Funktionsweise im Vordergrund, um eine solide Basis für das Verständnis der visuellen Datenverarbeitung zu legen. Diese Einführungsphase soll die Schülerinnen und Schüler mit den Möglichkeiten und Herausforderungen dieser Technologie vertraut machen.







Nach der Einführung in die Technologie folgt eine Phase, in der die Schülerinnen und Schüler ihre Programmierkenntnisse mit Hilfe von HuskyLens vertiefen. Die Projektvorschläge stellen nur eine kleine Auswahl an möglichen Unterrichtsideen dar.

Materialien

HuskyLens

Zugriff auf

[inf-schule | Informatiksysteme / Der Calliope mini als Informatiksystem / KI Kamera / Videoüberwachung mit KI](#)

KI-basierter Bildverarbeitungssensor (Zeitansatz: 10h)	
Kompetenzen	Anregungen und Hinweise
<p> angemessen auf Risiken bei der Nutzung von Informatiksystemen reagieren</p> <p> Wechselwirkungen zwischen Informatiksystemen und ihrer gesellschaftlichen Einbettung benennen</p> <p> sich weitere Informatiksysteme erschließen</p> <p> Kriterien zur Bewertung informatischer Sachverhalte anwenden</p> <p> Algorithmen entwerfen, implementieren und beurteilen</p> <p> kooperieren bei der Lösung informatischer Probleme</p>	<p>Wo begegnet uns Gesichtserkennung im Alltag?</p> <ul style="list-style-type: none"> ○ Fallbeispiele zu Authentifizierung oder Personenerkennung in Bildern – Smartphone, Türklingel, Spielekonsole, Smart-TV, Computer, Überwachungskamera ○ Vor- und Nachteile automatisierter Gesichtserkennung <ul style="list-style-type: none"> • Bequemlichkeit, Sicherheit oder Gefahr, Freiheit oder Kontrolle?, Probleme bei Datenschutz, Selbstbestimmung, Gefahr für Demokratie <p>Wir lernen einen KI-Bildverarbeitungssensor kennen.</p> <ul style="list-style-type: none"> ○ Wir untersuchen eine „Huskylens“: <ul style="list-style-type: none"> • Aufbau (Kamera, Display, Anschlüsse) ○ Experimente mit der KI-Bilderkennung <ul style="list-style-type: none"> • Kategorien: Gesichtserkennung, Objekterkennung, Linienverfolgung, Farberkennung und Erkennung von Tags (QR-Code) • Objekte anlernen / Gesichter erkennen (Objektorientierung, „Lernen“) <p>Wie „treffsicher“ ist die Bilderkennung?</p> <ul style="list-style-type: none"> ○ Fehlerquote bei der Bilderkennung: Erkennt mich die Gesichtserkennung immer? ○ „Überliste die KI!“ ○ Wovon könnte diese Fehlerquote abhängig sein? (Gesichtsausschnitt, Helligkeit, Entfernung, Perspektive, Brille, Maske, Haare im Gesicht, Bart / kein Bart ...)

Vorbereitung zur Programmierung mit Calliope

- Anschließen der HuskyLens an den Calliope (Kabel, Spannungsversorgung)
- Bibliothek in MakeCode einbinden
- Vorstellung wesentlicher Blöcke (Initialisierung ...)

Wir programmieren Anwendungen mit Hilfe des HuskyLens KI-Bildsensors:

- Realisierung von Projekten, wie z.B. Gesichtserkennung (Türöffner)
- Weitere Beispiele:
 - Linienverfolgung: Callibot
 - Farberkennung: Farbenspiel
 - Objekterkennung: Automatischer Torwart
- Gestenerkennung: Schnick-Schnack-Schnuck

Kontext „Die Welt im Modell“

Zielsetzung dieses Kontextes

In den Jahrgangsstufen 7/8 wurde mit Hilfe der Tabellenkalkulation eine einzelne Tabelle als erste strukturierte Speichermöglichkeit von Daten erarbeitet. Sortieren und Filtern der Daten ermöglichten es, aus größeren Datenmengen gezielt einzelne Datensätze abzufragen.







In diesem Kontext geht es jetzt um die Erweiterung der Möglichkeiten durch das Bilden von Beziehungen zwischen Tabellen, um einen komplexeren Ausschnitt aus einer „Miniwelt“ zu erfassen. In der Konsequenz müssen dann Verfahren erlernt werden, aus diesen miteinander verbundenen Tabellen Daten auszulesen; in diesem Kontext geschieht dies mittels der Abfragesprache SQL. Als Alternative oder Ergänzung sind grafische Werkzeuge denkbar.





Damit verfolgt der Kontext zwei Zielsetzungen: Das Erlernen der Grundstruktur einer Abfragesprache sowie die Abbildung einer Miniwelt in ein relationales Datenbankschema; dieses wird sukzessiv aufgebaut.

Materialien

Zugriff auf die Inhalte des Datenbankkapitels auf inf-schule und den dort integrierten Datenbanken


[inf-schule | Datenbanksysteme / Datenbank für Geodaten \(terra\)](#)


Die Welt im Modell (Zeitansatz: 12h)	
Kompetenzen	Anregungen und Hinweise
 bereits implementierte Systeme untersuchen  Sachverhalte analysieren und angemessene Modelle erarbeiten  die Darstellung und Strukturierung informatischer Sachverhalte begründen  Information in unterschiedlicher Form darstellen  arithmetische und logische Operationen kennen und verwenden  Strukturierungsmöglichkeiten von Daten zum Zusammenfassen gleichartiger und unterschiedlicher Elemente zu einer Einheit kennen und verwenden	<p>Die erste Datenbank</p> <ul style="list-style-type: none"> ○ Ausgehend von einer Liste von Orten mit ihren Kenndaten (Einwohnerzahl, Geokoordinaten) werden diese in einer Tabelle einer relationalen Datenbank gespeichert. <ul style="list-style-type: none"> • Die grundlegenden Begriffe „Datensatz“, „Attribut“ mit „Datentyp“ werden wiederholt. ○ In ersten einfachen Anfragen werden die Grundstruktur der Abfragesprache SQL erarbeitet. <ul style="list-style-type: none"> • SELECT – Auswahl der anzuzeigenden Attribute • FROM – Auswahl der abgefragten Tabelle • WHERE – Bedingungen auf den Datensätzen • ORDER BY – Sortieranweisungen ○ Die Abfragesprache erlaubt nun, aus der großen Datenmenge gezielt Daten zu selektieren und neue Fragestellungen auszuwerten. ○ In einem ersten Abgleich wird das Modell mit der realen Welt verglichen. <ul style="list-style-type: none"> • Diskussion der (vorgegebenen) Wahl der Attribute und ihrer Datentypen – welche Einschränkungen hat schon das einfache Modell? <p>↳ <i>Miniwelt, Abfrage</i></p>


-  die Verwaltung und Speicherung großer Datenmengen mithilfe eines Datenmodells modellieren
-  einfache Datenmodelle in relationale Modelle umsetzen und diese mit einem Datenbanksystem realisieren
-  netzartige Strukturen erstellen
-  grundlegende Operationen zum Zugriff auf die Bestandteile strukturierter Daten kennen und verwenden


Daten verknüpfen

- Es ist gar nicht so einfach, bei einer großen Datenmenge einen bestimmten Datensatz zu finden (z.B. die Hauptstädte Berlin und Washington).
 - Notwendigkeit, ein eindeutiges Identifizierungsmerkmal zu finden bzw. künstlich zu erzeugen, den „Primärschlüssel“
 - ▶ Parallelen zu anderen „Primärschlüsseln“ aus der realen Welt herstellen (Ausweisnummer, ISBN, etc.)
- Fehler in den Daten zeigen das Problem bei der mehrfachen („redundanten“) Speicherung der gleichen Daten.
 - Problem, dass mehrere unterschiedliche Objekte (Länder und Kontinente) in einer Tabelle gespeichert sind
 - Notwendigkeit, die Daten in mehrere Tabellen aufzuteilen
- Die nun über mehrere Tabellen verteilten Daten müssen verbunden werden.
 - Über die Schlüssel (Fremdschlüssel → Primärschlüssel) wird die Verbindung der Daten wieder hergestellt.
 - ▶ Visualisierung der Verbindung durch Pfeile zwischen den Datensätzen
 - ▶ Visualisierung der Verbindung auf Attributebene durch Pfeile zwischen den Attributen der Tabelle
- Für Abfragen muss die Syntax von SQL erweitert werden, so dass Daten mehrerer Tabellen abgefragt werden können.
 - Syntax in SQL auf mehrere Tabellen erweitern (FROM-Teil und Punktnotation für die Attribute)
 - Joinbedingung zur Auswahl der verknüpften Attribute
 - ↳ *Primärschlüssel, Fremdschlüssel, (Objekt-)Beziehung*

 Diagramme, Grafiken und Modelle nutzen, um sich informatische Sachverhalte selbstständig zu erarbeiten

 Diagramme, Grafiken sowie Ergebnisdaten interpretieren

 Diagramme und Grafiken gestalten, um informatische Sachverhalte zu beschreiben und mit anderen darüber zu kommunizieren

 Vorgehensweisen bei der Modellierung informatischer Sachverhalte begründen

Komplexere Beziehungen

- Die Modellierung über direkte Fremdschlüsselbeziehungen reicht für bestimmte Situationen nicht aus (Beispiel der Flüsse und der Orte, die an diesen liegen).
 - Einfache Fremdschlüssel können nicht verwendet werden (keine „Listen“ in den Schlüsseln erlaubt, da nicht atomar)
 - Beziehungstabelle, die zunächst nur aus Schlüsseln besteht, als Lösung

optional:

- Klassifizierung von Beziehungstypen (n:m-Beziehung, 1:n-Beziehung)
- Visualisierung als ER-Diagramm
- Neue Daten durch Aggregation gewinnen
 - Erweitern der SQL-Syntax um Aggregationsfunktionen (vor allem COUNT(*)), um Gesamtzahlen zu ermitteln.
 - In einem zweiten Schritt das Bilden von Gruppen innerhalb der Daten mittels der GROUP BY-Anweisung.

Grenzen des Modells und seine Erweiterung

- Einfache Erweiterungen des Modells der Geo-Datenbank vertiefen die zuvor erlernten Aspekte – insbesondere die Verknüpfung von Daten.
 - Anlegen von Ozeanen, Landesteile, etc.
- Weiterführende Beispiele lassen die Grenzen eines Modells erkennen.
 - Sonderfälle wie z.B. mehrere Hauptstädte
 - ▶ Diskussion, inwieweit jeder Sonderfall in einem Modell abgebildet werden muss
 - ▶ Diskussion um die Konsequenzen (z.B. für Abfragen sowie die Datenpflege), wenn ein Modell dadurch komplexer wird

Kontext „Spacebug – Algorithmen für das Weltall“

Zielsetzung dieses Kontextes

Die Schülerinnen und Schüler sollen Algorithmen textbasiert in Python implementieren. Sie verwenden dabei die gängigen Kontrollstrukturen Sequenz, Fallunterscheidung und Wiederholung. Zudem setzen sie Variablen ein und strukturieren ihre Programme durch einfache Unterprogramme.

Durch Verwendung von Kommentaren, sinnvollen Bezeichnen und einheitlicher Einrückung wird dabei eine gute Lesbarkeit der Quelltexte gewährleistet.

Mit der Programmierumgebung Spacebug bekommen die Schülerinnen und Schüler direkte visuelle Rückmeldung über die Richtigkeit ihrer Programme. Zudem steht ein Debugger zur Verfügung und es besteht die Möglichkeit die Veränderung von Variablenwerten direkt zu beobachten.

In der Regel sind bei den Beispielaufgaben bereits mehrere Szenarien („Testwelten“) vorhanden, mit der die Allgemeingültigkeit der Programme überprüft werden kann.

Voraussetzungen der Schülerinnen und Schüler








Dieser Kontext greift verschiedene Kompetenzen aus dem Inhaltsbereich Algorithmen der Jahrgangsstufen 5 bis 8 erneut auf.





Materialien

Zugriff auf

[inf-schule | Imperative Programmierung / Python Spacebug Online](#)

Spacebug – Algorithmen für das Weltall (Zeitansatz 16h)

Kompetenzen	Anregungen und Hinweise
 einfache Algorithmen entwerfen und testen  Algorithmen entwerfen, implementieren und beurteilen  Quelltexte von Programmen nach Vorgaben modifizieren und ergänzen  Problemlösungen in einer Programmiersprache angeben  bei der Implementierung die algorithmischen Grundbausteine verwenden  Sachverhalte durch zweckdienliches Zerlegen und Anordnen strukturieren  Arbeitsabläufe und Handlungsfolgen planen	<p>Wie kann ich die Argo steuern?</p> <ul style="list-style-type: none"> ○ Kennenlernen des Tools (Programmfenster, Weltfenster, Bedienung) ○ Schrittweise Ausführung der Elementaranweisungen (laufen, drehen, legen, aufheben) ○ Textuelle Programmierung <p>Kontrollstrukturen in Python</p> <ul style="list-style-type: none"> ○ Argo bekommt verschiedene „Missionen“, die dazu führen, dass nach und nach unterschiedliche Kontrollstrukturen benötigt werden (Sequenzen, Wiederholungen, Fallunterscheidungen inkl. jeweils benötigter Bedingungen) Dadurch ergibt sich <ul style="list-style-type: none"> • Problemorientierter Zugang • Rolle der Einrückung (aus syntaktischen Gründen und zur besseren Übersicht) • Fehlerbehandlung (Syntax, Semantik, Debugging) • Testen der Programme an verschiedenen Weltszenarien • Vergleich von Darstellungsformen: textuelle Programmierung ↔ blockbasierte Programmierung ↔ Flussdiagramm <p>Wie kann ich ein Programm übersichtlicher gestalten?</p> <ul style="list-style-type: none"> ○ Verwendung von einfachen Unterprogrammen (s. inf-schule) ○ Kommentare zur Strukturierung

 Variablen und Wertzuweisungen verwenden  Fehlermeldungen bei der Arbeit mit Informatiksystemen interpretieren und produktiv nutzen  das Modellverhalten durch zielgerichtete Änderungen beeinflussen  bei der Lösung informatischer Probleme kooperieren	<h3>Variablen mit Python</h3> <ul style="list-style-type: none">○ Weitere Missionen führen zu einfachen Zählvariablen, die mit sprechenden Bezeichnern dargestellt werden.○ Dabei verstehen wir (weiterhin) Variablen (nur) als Platzhalter für einen veränderbaren Wert○ Thematisieren, dass Rückgabewerte von Sensoren (booleschen) Wahrheitswerten entsprechen<ul style="list-style-type: none">▶ Debugging mit Anzeige der Variablenwerte (z.B. als Trace-Tabelle)○ optional:<ul style="list-style-type: none">• weitere Datentypen, z.B. Zeichenketten• weitere Vertiefungen im Spacebug-Kapitel auf inf-schule
--	--