

Ein Gesamtkonzept für die Schulinformatik in Bayern (Hubwieser, 1999)

Das Fundamentum

Alter: 5./6. JGSt. (10-11 Jahre)

Ziel: Grundlegende Einführung in die wichtigsten Begriffe und Denkweisen der Informatik

Vorgehen: altersgemäß spielerisch und handlungsorientiert, jedoch keinesfalls ungenau oder unsystematisch.

Themenschwerpunkt: Arbeit mit **Dokumenten**

Lerninhalte:

1. Erstellung und Präsentation von Dokumenten

Textverarbeitung:

Texte, Absätze, Zeichen als Objekte mit Attributen. Markieren, Ausschneiden, Einfügen als grundlegende Operationen auf Texten, Einbau von Grafiken.

Hypertext:

Struktur von Hypertextdokumenten, Einbau von Verweisen und multimedialen Objekten

Tabellenkalkulation:

Zeilen, Spalten und Zellen, Typen von Zellinhalten, Verweise auf andere Zellen, einfache Rechenfunktionen, einfache Präsentationstechniken

Grafikprogramme:

typische Datenstrukturen von Vektor- und Pixelgrafik, mögliche Operationen darauf, Abschätzung des Platzverbrauchs, Manipulationsmöglichkeiten.

Hierarchische Dateisysteme:

Dateien als Container für die Daten von Standardsoftwareprogrammen, Löschen und Kopieren von Dateien.

2. Verwaltung und Transport von Dokumenten

Ordner zur hierarchischen Verwaltung von Dateien, beispielhafte Ordnersysteme, Verschieben und Kopieren von Dateien und Unterordnern, Versand von Dokumenten mit Hilfe von elektronischer Post, Publikation von Dokumenten im Internet, Client-Server Prinzip, Systematische Suche nach Dokumenten in Verzeichnisstrukturen und im Internet

3. Verarbeitung von Dokumenten und Automatisierung dieser Verarbeitung

Umgangssprachliche Formulierung von Verarbeitungsvorschriften für Objekte aus dem Bereich der Standardsoftware

Bausteine von Algorithmen: Sequenz, Wiederholung und Auswahl

Funktionsweise von Rechenanlagen in einfachen Blockschaltbildern

Automatisierung von Verarbeitungsvorgängen durch einfache Makros

Unterscheidung von Daten- und Programmdateien.

Die Aufbaumodule

Alter: 7. - 9. JGSt. (13-16 Jahre)

Ziel: Sicherung der Kontinuität bei der Beschäftigung mit der Informatik, Vertiefung der eingeführte Begriffe und Denkweisen

Themenschwerpunkt: Dokumente und Verarbeitungstechniken.

Lerninhalte:

Textverarbeitung:

Einbindung von Grafiken, Tabellen und anderen Objekten in Texte, Layoutprinzipien, Layoutvorlagen, Serienbriefe, Grundprinzipien von Texterkennungssystemen.

Hypertextstrukturen:

Abbildung logischer Beziehungsgefüge in Dokumentenstrukturen, Anreicherung mit multimedialen oder aktiven Elementen, Präsentation durch Webserver.

Kommunikation in Rechnernetzen:

Dienste in Rechnernetzen (Vertiefung), Informationssuche im Internet, Versand und Herunterladen von Dokumenten, Produktion und Veröffentlichung eigener Dokumente.

Tabellenkalkulation:

Berechnungen und Präsentation mit Hilfe von Tabellenkalkulation, Funktionsbegriff, Verknüpfung von Funktionen, ausführliche Gestaltung von Präsentationsgraphiken, Einsatz von Tabellenkalkulationen im Bereich der Wirtschaft

Computergrafik:

Pixel- und Vektorgrafiken, Abschätzung des Platzbedarfs, Scannerprinzip, mathematische Formulierung von ebenen oder räumlichen Transformationen, Grundstrukturen von CAD-Systemen.

Konzeption von Rechenanlagen:

Aufbau und funktionsweise von Computern und Computernetzen, Konzeption und Planung von lokalen Netzwerken, Rechte- und Verzeichnisstrukturen, Sicherheitsstrategien

Selbstverständlich ist auch eine Vermischung einzelner Module denkbar.

In diesem Bereich der Informatik sollten die Schülerinteressen auf jeden Fall Vorrang haben.

Allgemeinbildungsmodule

Alter: 10. /11. JGSt. (16-17 Jahre)

Ziel: *vertiefte informatische Allgemeinbildung* zur Vorbereitung auf Studium und Beruf

Themenschwerpunkt: Modellierung von realen oder geplanten Systemen

Lerninhalte:

1. **Prinzipien der Repräsentation** von Information (ca. 16 Std.)

Grundprinzip der Informationsverarbeitung:

Aufbereiten und Darstellen von Informationen, Verarbeiten, Transportieren, Rekonstruieren und Interpretieren von Repräsentationen

Modellierung als idealisierte, vereinfachte Beschreibung eines realen oder geplanten

Systems

Erarbeitung einfacher Beschreibungstechniken

an Beispielen aus der Erfahrungswelt der Schüler: Datenstrukturen (Texte, Tabellen, Grafiken, Dateien und Verzeichnisse), Verarbeitungsvorschriften (Algorithmen, Kontrollstrukturen, Struktogramme), Systembeschreibungen (Datenflußdiagramme, Zustands- Übergangsdigramme)

Repräsentation durch Sprache:

Zeichensatz, Syntax, Semantik, Unterschiede zwischen natürlichen und künstlichen Sprachen,

2. Datenmodellierung (ca. 20 Std.)

Modellierung mit Hilfe von Entity-Relationship-Diagrammen:

Entitäten (Datenobjekte); Klassen von Entitäten; Attribute und Wertebereiche; Beziehungen zwischen Entitätsklassen

Umsetzung von ER-Modellen in relationale Datenbanken:

Schema, Tabellen, Spalten, Datentypen, Datensätze, Schlüssel, Redundanz, Konsistenz, Normalformen 1-3 · Operationen auf Tabellen: Datenmanipulation, Abfragen, Berichte, Datenschutz, Client-Server Prinzip.

3. Zustandsorientierte Modellierung und Algorithmen (ca. 20 Std.)

Modellierung von Abläufen mit Hilfe von Zustandsübergangsdigrammen:

Zustände und

Übergänge; Andauernde, auslösende und ausgelöste Aktionen; endliche deterministische Automaten mit und ohne Ausgabe

Implementierung von Automaten durch imperative Programme: B

austeine von Algorithmen (Zuweisung, Kontrollstrukturen: Sequenz, Auswahl, Wiederholung),

Darstellung durch Struktogramme, Zustände von Variablen, Steuerung der Übergänge durch Kontrollstrukturen, Syntaxdiagramm

Automatenmodell des von Neumann-Rechners:

Realisierung von Variablen und Verarbeitungsvorschriften durch Zustände von Speicherzellen

4. Funktionale Modellierung (ca. 12 Std.)

Aufteilung komplexer Systeme in Teilsysteme (Komponenten), Datenflußdiagramme, schrittweise Verfeinerung

Modularisierung von imperativen Programmen:

Funktionen und Prozeduren;

Schnittstellen; Ein- und Ausgabeparameter; globale und lokale Variable; strukturierte Programmierung

Rekursion als Hilfsmittel zur Problemlösung:

Rekursive Beschreibung von Algorithmen aus funktionaler Sicht

5. Objektorientierte Modellierung (ca. 16 Std.)

Objektmodell:

Identifikation von Objekten, Klassenbildung

Objekte als Automaten:

Objekte vereinen Datenstrukturen (Attribute) und Verarbeitungsvorschriften (Methoden); Attributname und –wert; Objekt als Instanz seiner Klasse; Datenkapselung

Kommunikation zwischen Objekten:

Gegenseitiger Aufruf von Methoden durch Botschaften

6. Verteilte Systeme (ca. 8 Std.)

Modellierung paralleler Abläufe mit Hilfe von einfachen Message Sequence Charts

Kopplung paralleler Prozesse; Zugriff auf gemeinsame Ressourcen

Kommunikation zwischen parallelen Prozessen:

Protokolle, Protokollschichten

Ein Beispiel für ein verteiltes System mit parallelen Prozessen (**Internet**)

7. Unterrichtsprojekt:

Entwicklung eines hinreichend komplexen Systems (ca. 20 Std.)

Phasenmodell der Softwareentwicklung:

Planung, Analyse, Entwurf, Implementierung, Test und Einführung, Wartung und Pflege

Anwendung des Phasenmodells bei der Konstruktion und Simulation eines komplexen Systems zu einer selbstgewählten Aufgabenstellung; Beachtung von Prinzipien des Datenschutzes und Datensicherheit; Simultaner Einsatz mehrerer Modellierungstechniken während verschiedener Phasen des Projektes

Erstellung einer ausführlichen Dokumentation von Vorgaben, Projektverlauf und Ergebnissen der einzelnen Phasen; ausführliche Abschlußphase mit kritischer Wertung des gesamten Projektverlaufs

Oberstufe

Alter: 11./12. JGSt. (17-18 Jahre)

Ziel: Spezielle Studienvorbereitung (*Fachwissenschaftspropädeutik*)

Themenschwerpunkt: Vertiefung der Modellierung, Aspekte der theoretischen Fundierung und der Implementierung

Lerninhalte:

Formale Sprachen:

Notationen: Grammatiken, BNF-Notation, Syntax-Diagramme, Programmiersprachen, endliche Automaten, Kellerautomaten

Rekursive Datenstrukturen:

Verwendung von Listen und Bäumen zur Verarbeitung von Information, Funktionale Programmierung

Objektorientierung (Fortsetzung):

Höhere Konzepte der Objektorientierung: Vererbung, Polymorphismus, Vertiefung der Objektorientierten Programmierung

Rechnersysteme:

Einführung in die Rechnerarchitektur, Grundlegende Konzepte von Betriebssystemen

Konzepte von Rechnernetzen: Adressierungs- und Vermittlungstechniken

Einführung in die Aussagen- und Prädikatenlogik:

Modellierung mit Hilfe von Regeln, Logikprogrammierung

Berechenbarkeit und Komplexität:

Abstrakte Maschinenmodelle: Turingmaschine,

Registermaschine, Existenz nichtberechenbarer Funktionen, Komplexitätsklassen

Projektarbeit (Fortsetzung): Schwerpunkt Zusammenarbeit von Projektgruppen