

# **Beispiele für schulinterne Fachpläne**

## **Informatik Sekundarstufe II**

## **Impressum**

### **Herausgeber:**

Landesinstitut für Schule und Medien Berlin-Brandenburg (LISUM)  
14974 Ludwigfelde-Struveshof

Tel.: 03378 209-200

Fax: 03378 209-232

Internet: [www.lisum.berlin-brandenburg.de](http://www.lisum.berlin-brandenburg.de)

### **Autorinnen und Autoren:**

Roland Ebner,  
Friedrich-Schiller-Gymnasium, Königs-Wusterhausen, Brandenburg

Dr. Jan Hartmann,  
Fachdezernent für Informatik, Landesinstitut für Schule und Ausbildung,  
Mecklenburg-Vorpommern

Thomas Lösler,  
Philipp-Melanchthon-Gymnasium, Herzberg/E., Brandenburg

### **Ansprechpartner im LISUM:**

Dr. Peter M. Schulze

### **Grafiken:**

Alle Abbildungsrechte für Fotos und Grafiken liegen bei den Autoren.

© Landesinstitut für Schule und Medien Berlin-Brandenburg (LISUM); März 2008

Dieses Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Alle Rechte einschließlich Übersetzung, Nachdruck und Vervielfältigung des Werkes vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Landesinstituts für Schule und Medien Berlin-Brandenburg in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren) reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden. Eine Vervielfältigung für schulische Zwecke ist erwünscht. Das LISUM ist eine gemeinsame Einrichtung der Länder Berlin und Brandenburg im Geschäftsbereich des Ministeriums für Bildung, Jugend und Sport des Landes Brandenburg (MBJS).

## Inhaltsverzeichnis

Vorwort .....	4
1 Rechner und Netze (GK/LK).....	5
2 Softwareentwicklung (GK/LK).....	11
3 Sprachen und Automaten (LK) .....	23
4 Sprachen und Automaten (GK) .....	30

## Vorwort

Das vorliegende Material zum Rahmenlehrplan Informatik der Sekundarstufe II bietet eine erste Orientierung zum Kerncurriculum und seine Einordnung in die schulinternen Fachpläne an. Ausgehend von den Eingangsvoraussetzungen wird eine Brücke geschlagen zu den abschlussorientierten Standards und einem möglichen Abitur im Fach Informatik auf der Grundlage der Einheitlichen Prüfungsanforderungen der Kultusministerkonferenz.

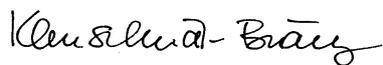
Die Entwicklung des vorliegenden Materials wurde von der gemeinsamen Rahmenplan-Kommission Informatik der Länder Berlin, Brandenburg und Mecklenburg-Vorpommern bereits während der Erarbeitung des Rahmenlehrplanes begonnen. Nach der Fertigstellung des Plans wurde der bis dahin vorliegende Entwurf auf Grund der Anregungen von den Autoren noch einmal überarbeitet. Insbesondere wurden die im Unterricht erprobten Aufgaben zum Teil umformuliert.

Das vorliegende Material ergänzt die Anregungen zum Rahmenlehrplan Informatik in der Sekundarstufe II um Beispiele zur Gestaltung schulinterner Fachpläne. Die Anregung zur Diskussion und zur Entwicklung eigener Varianten in den Fachkonferenzen ist ein wesentliches Ziel dieses Materials. Die Vorschläge wurden an ausgewählten Stellen um beispielhafte Aufgaben ergänzt.

Wir danken den Kolleginnen und Kollegen, die mit ihren Hinweisen und Kritiken zur Verbesserung der Aufgabenbeispiele beigetragen haben.

Anfragen, Meinungen, Kritik und ergänzende Beispiele sind ausdrücklich erwünscht.

Viel Erfolg bei der Umsetzung der Anregungen.



Mascha Kleinschmidt-Bräutigam

Leiterin der Abteilung Unterrichtsentwicklung

# 1 Rechner und Netze (GK/LK)

	<b>Eingangsvoraussetzungen Die Schülerinnen und Schüler</b>	<b>Abschlussorientierte Standards Die Schülerinnen und Schüler</b>
<b>Informatisches Modellieren</b>  <i>Modelle erstellen und bewerten</i>	<ul style="list-style-type: none"> <li>• beschreiben Modelle als vereinfachtes Abbild der realen Welt               <ul style="list-style-type: none"> <li>○ nutzen ein einfaches 3-Schichten-Modell zur Beschreibung der Kommunikation in Netzen</li> </ul> </li> <li>• hinterfragen und bewerten Ergebnisse einer Modellbildung kritisch               <ul style="list-style-type: none"> <li>○ erkennen die Grenzen des 3-Schichten-Modells</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• variieren und erweitern vorgegebene Modelle               <ul style="list-style-type: none"> <li>○ erkennen Grenzen des 3-Schichten-Modells und passen die Modellierung auf die Netzwerkkommunikation an (4 Schichten TCP-Modell oder 7 Schichten OSI-Modell)</li> </ul> </li> <li>• reflektieren und beurteilen die eigene Modellierung</li> </ul>
<b>Mit Information umgehen</b>  <i>Information in Form von Daten darstellen und verarbeiten</i>	<ul style="list-style-type: none"> <li>• unterscheiden zwischen Information und Daten               <ul style="list-style-type: none"> <li>○ wissen, dass Information in Daten repräsentiert und in Datenpaketen im Netzwerk übertragen werden</li> <li>○ Aufbau der Datenpakete ist genormt, Norm ist in Protokollen festgelegt</li> </ul> </li> <li>• nutzen zielgerichtet bereitgestellte Informationssysteme, digitale Datenbestände und Datenbanken               <ul style="list-style-type: none"> <li>○ nutzen zielgerichtet Suchmaschinen und digitale Datenbestände (z. B. Wikipedia)</li> </ul> </li> </ul>	
<b>Informatiksysteme verstehen</b>  <i>Wirkprinzipien kennen und anwenden</i>	<ul style="list-style-type: none"> <li>• beschreiben Grundlagen des Aufbaus und der Arbeitsweise eines Informatiksystems               <ul style="list-style-type: none"> <li>○ beschreiben den Aufbau eines Informatiksystems aus den grundlegenden Bestandteilen (Hardware, Software, Vernetzung)</li> </ul> </li> <li>• beschreiben die Grundlagen der Rechnerkommunikation in lokalen Netzwerken.               <ul style="list-style-type: none"> <li>○ Beschreiben Rechnerkommunikation als Kommunikation in Schichten (Anwendung, Übersetzung, Übertragung) auf der Grundlage von Protokollen (http, ip, Ethernet)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• diskutieren Funktionalität, Leistungsfähigkeit und Zuverlässigkeit von Informatiksystemen               <ul style="list-style-type: none"> <li>○ werten die Bandbreite der Datenübertragung als ein Kriterium der Leistungsfähigkeit von Netzen</li> <li>○ diskutieren die Ausfallsicherheit als ein Kriterium der Zuverlässigkeit von Informatiksystemen</li> </ul> </li> <li>• erläutern einfache Schichtenmodelle von Netzwerken und Informatiksystemen.               <ul style="list-style-type: none"> <li>○ kennen die Adressierung auf den einzelnen Schichten bei der Datenkapselung (Mac-Adresse, IP-Adresse, Port-Adresse)</li> </ul> </li> </ul>
<b>Problemlösen</b>  <i>Probleme erfassen und mit Informatiksystemen lösen</i>	<ul style="list-style-type: none"> <li>• beschreiben algorithmische Abläufe umgangssprachlich und graphisch               <ul style="list-style-type: none"> <li>○ beschreiben die Abläufe bei der Kommunikation in Netzen umgangssprachlich</li> </ul> </li> </ul>	

	<b>Eingangsvoraussetzungen Die Schülerinnen und Schüler</b>	<b>Abschlussorientierte Standards Die Schülerinnen und Schüler</b>
<b>Kommunizieren und Kooperieren</b>  <i>Teamarbeit organisieren und koordinieren</i>	<ul style="list-style-type: none"> <li>• nutzen Rechnernetzwerke zur Kommunikation</li> <li>• verwenden im angemessenen Rahmen die Fachsprache</li> <li>• dokumentieren und präsentieren ihre Arbeitsergebnisse.</li> </ul>	<ul style="list-style-type: none"> <li>• verfügen über eine angemessene Fachsprache und verwenden sie sachgerecht</li> <li>• verwenden selbstständig Fachtexte, Dokumentationen und Hilfesysteme</li> <li>• setzen netzbasierte Kommunikations- und Kooperationssysteme in der Gruppenarbeit ein</li> <li>• beachten die Netiquette bei der Kommunikation</li> <li>• dokumentieren, visualisieren, präsentieren und verteidigen Ergebnisse der Teamarbeit</li> <li>• erfassen, reflektieren und diskutieren informatische Sachverhalte aus nicht didaktisch aufbereiteten authentischen Texten (z. B. Presseartikel).</li> </ul>
<b>Wechselwirkungen zwischen Informatiksystemen, Mensch und Gesellschaft beurteilen</b>  <i>Anwendungen erfassen und Auswirkungen abschätzen</i>	<ul style="list-style-type: none"> <li>• beurteilen die historische Entwicklung der Informatik vor dem Hintergrund gesellschaftlicher Interessen und technischer Entwicklungen <ul style="list-style-type: none"> <li>○ beurteilen den Erfolg des WWW unter Berücksichtigung von Nutzeranforderungen</li> </ul> </li> <li>• analysieren anhand von Fallbeispielen Probleme des Persönlichkeits- und Datenschutzes sowie der Datensicherheit, beachten Urheberrechte. <ul style="list-style-type: none"> <li>○ Beispiel Kreditkarteninformationen: Vernetzung als Grundlage der Zusammenführung von örtlich getrennten Daten zur Erstellung eines Bewegungsprofils von Personen</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• bewerten Risiken und Chancen von Informatiksystemen</li> <li>• nehmen das Recht auf informationelle Selbstbestimmung wahr und halten die Gesetze zum Datenschutz ein</li> <li>• bewerten Probleme der Mensch-Maschine-Kommunikation und der Ergonomie</li> <li>• analysieren politische und gesellschaftliche Rahmenbedingungen wichtiger informatischer Entwicklungen und beurteilen deren Wirkungen.</li> </ul>
		<ul style="list-style-type: none"> <li>• nur LK: beurteilen die Grenzen des Einsatzes von Informatiksystemen aufgrund individueller und gesellschaftlicher Verantwortung.</li> </ul>

## Zeitlicher Verlauf und methodische Hinweise

Schwerpunkte der UE	Inhalte/Begriffe	methodisch/didaktische Hinweise	Kontext (Erfahrungswelt der Schüler, weitere Themen, andere Fächer)
Einführung Einfaches Schichtenmodell	<ul style="list-style-type: none"> <li>Analysieren der Briefkommunikation mit dem Ziel, gleiche Schichten und Vereinbarungen zu identifizieren</li> <li>Anwenden des Schichtenmodells auf die Morsekommunikation.</li> <li>Übertragen dieser Ergebnisse auf die Kommunikation in Netzwerken.</li> </ul>	<ul style="list-style-type: none"> <li>Einführungsbeispiel Brief-Kommunikation zw. deutschem und chinesischem Geschäftsführer (und/oder Kommunikation per Morsealphabet zweier Kapitäne) wird auf Kommunikation in Netzwerken überführt.</li> <li>Einfache Schichten und Regeln (Protokoll) werden herausgearbeitet.</li> <li>Eine wesentliche Protokollvereinbarung ist Kennzeichnung Adressat und Absender.</li> </ul>	<ul style="list-style-type: none"> <li>Arbeitsblatt zu den einfachen Schichtenmodellen (Kommunikation)</li> <li>Arbeitsblatt zu modernen Kommunikationsmitteln (Methode, Hilfsmittel zum Transport, Übertragungsmedium, Kennzeichnung des Absenders, Kennzeichnung des Empfängers)</li> </ul>
IP- und MAC-Adressierung	<ul style="list-style-type: none"> <li>Benutzen des lokalen Netzes zur Kommunikation.</li> <li>Analysieren der Topologie und der benutzten Adressen.</li> <li>Einbindung eines weiteren PC in das vorhandene Netz.</li> </ul>	<ul style="list-style-type: none"> <li>Benutzen-Analysieren-Gestalten</li> <li>Vereinbarungen (Physikalische Adresse, IP-Adresse, Subnetz-Maske, Standardgateway ...)</li> <li>Welche Einstellungen (Vereinbarungen) müssen gleich sein und welche müssen sich ändern, wenn ein weiterer PC ins Netz integriert wird?</li> <li>Topologieerweiterung: LV zu Standardgateway und Subnetzbildung</li> </ul>	<ul style="list-style-type: none"> <li>Mit ipconfig /all wird der eigene Arbeitsplatz analysiert</li> <li>Mit ping die Erreichbarkeit anderer PC im Netz testen</li> <li>Schülervortrag IP-Adressierung</li> <li>Nach LV trace Befehl anwenden um Bedeutung des Gateway zu erschließen</li> </ul>
Vernetzungspraxis	<ul style="list-style-type: none"> <li>Anwenden der Kenntnisse über IP-Adressierung und Subnetzbildung</li> </ul>	<ol style="list-style-type: none"> <li>Topologie und Adressräume <ul style="list-style-type: none"> <li>Vernetzen zweier PC mit crossover-Kabel, über Switch und Router</li> <li>Verbindung der geschichteten Stationen per Router</li> <li>Test mit ping / trace</li> </ul> </li> <li>Client-Server-Prinzip <ul style="list-style-type: none"> <li>Webserver</li> <li>FTP-Server</li> <li>Mailserver</li> </ul> </li> </ol>	Stationsbetrieb organisieren <ol style="list-style-type: none"> <li>PC-PC</li> <li>PC-PC-Drucker</li> <li>PC-PC-Netzwerkdrucker</li> <li>PC-Switch-PC</li> <li>Aufgabe zum Subnetting</li> <li>Webserver</li> <li>Mailserver</li> </ol>

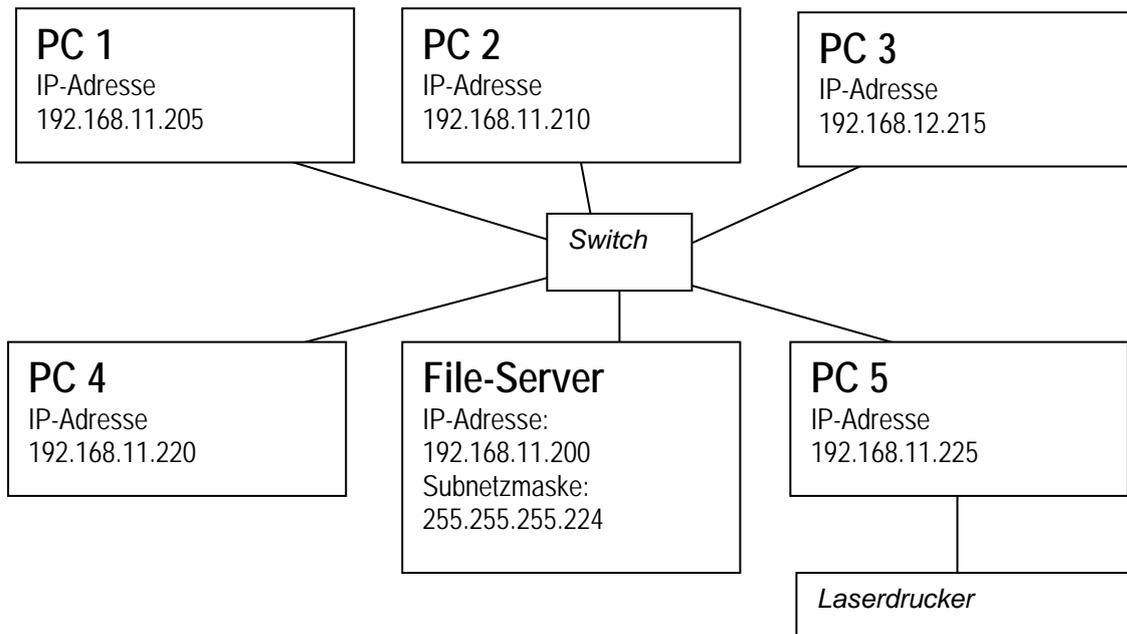
<b>Schwerpunkte der UE</b>	<b>Inhalte/Begriffe</b>	<b>methodisch/didaktische Hinweise</b>	<b>Kontext (Erfahrungswelt der Schüler, weitere Themen, andere Fächer)</b>
Client-Server am Beispiel der E-Mail-kommunikation	<ul style="list-style-type: none"> <li>• Benutzung des E-Mail-Dienstes des Internets.</li> <li>• Beschreiben von Anforderungen aus Sicht des Nutzers.</li> <li>• Prinzipielle Funktionsweise der Kommunikation per E-Mail.</li> </ul>	<ul style="list-style-type: none"> <li>• Was muss aus Sicht des Nutzers eine E-Mail-Kommunikation leisten und welche Probleme und Gefahren müssen ausgeschlossen werden?</li> <li>• Analyse des Headers einer echten Mail lässt Rückschlüsse auf die tatsächlichen Vereinbarungen zu.</li> <li>• Diskussion von Vertraulichkeit, Authentizität und Datensicherheit, aber auch Missbrauch von Dateianhängen</li> <li>• Netiquette</li> </ul>	<ul style="list-style-type: none"> <li>• Festlegen von Vereinbarungen mit dem Ziel ein Protokoll für das Versenden und das Empfangen von E-Mail zu beschreiben</li> <li>• Einbindung eines E-Mail-servers in das einfache Schichtenmodell der Kommunikation zweier Personen.</li> </ul>
Rechner und Netze	<ul style="list-style-type: none"> <li>• Kenntnisse der Schichtenarchitektur und Protokolle auf Rechner und Betriebssysteme übertragen.</li> <li>• Notwendige Vereinbarungen für die Kommunikation der einzelnen Komponenten eines Von-Neumann-Rechners.</li> </ul>	<ul style="list-style-type: none"> <li>• Einstieg mit Schülervorträgen.</li> <li>• Analyse in Bezug auf Schichtenarchitektur und Protokolle.</li> </ul>	<ul style="list-style-type: none"> <li>• Welche Gemeinsamkeiten lassen sich bei Rechnern und Netzwerken finden?</li> <li>• BUS-System, Analogie zur Schicht 1</li> </ul>
Informatiksysteme und Gesellschaft	<ul style="list-style-type: none"> <li>• Gesellschaftliche Bedeutung von Vernetzung (und Kommunikation).</li> <li>• Diskutieren ethische und moralische Grenzen der technischen Entwicklungen.</li> </ul>	<ul style="list-style-type: none"> <li>• Bedeutung der Vernetzung und der Kommunikation für die wirtschaftliche Entwicklung diskutieren</li> <li>• Welche Schritte sind notwendig, um ein Dritte-Welt-Land an der internationalen Kommunikation teilhaben zu lassen? Wie können die notwendigen Investitionen finanziert werden?</li> <li>• Jeder PC ist mit einer IP-Adresse weltweit eindeutig adressierbar. Sollte nicht jeder Mensch eine eindeutige (IP-)Adresse besitzen, mit der er sich an jedem Rechner der Welt authentifizieren kann?</li> </ul>	<ul style="list-style-type: none"> <li>• Welche Einschränkungen würde es für die Gesellschaft geben, wenn keine Vernetzung der PC existieren würde?</li> </ul>

## Aufgaben zum Themenfeld Rechner und Netze

Das Aufgabenfeld leistet einen Beitrag zum Kompetenzbereich Informatiksysteme verstehen, Wirkprinzipien kennen und anwenden. Darüber hinaus bietet es Möglichkeiten Wechselwirkungen zwischen Informatiksystemen, Mensch und Gesellschaft zu beurteilen und zu diskutieren sowie Grenzen des Einsatzes von Informatiksystemen aufzuzeigen.

Die folgenden Aufgaben stammen aus dem schriftlichen Leistungskurs-Abitur Informatik M-V 2004:

Bei der Modernisierung einer Schule wird auch die Vernetzung der PC-Kabinette und der Schulverwaltungsarbeitsplätze neu strukturiert. Die fünf lokalen Arbeitsstationen der Verwaltung werden mit einem Server und einem netzwerkfähigen Laserdrucker vernetzt. Ein Azubi erstellt einen Vorschlag für das Verwaltungsnetz.



Aufgabe	Abschlussorientierter Standard
<ul style="list-style-type: none"> <li>Diskutieren Sie den gegebenen Vorschlag und korrigieren Sie vorhandene Fehler.</li> </ul>	<ul style="list-style-type: none"> <li>Erfassen, reflektieren und diskutieren informatische Sachverhalte aus nicht didaktisch aufbereiteten authentischen Texten.</li> <li>Wenden das Adressierungsprinzip (IP-Adresse, Subnetze) in Netzwerken auf der Basis des Internetprotokolls an.</li> <li>Diskutieren Funktionalität, Leistungsfähigkeit und Zuverlässigkeit von Informatiksystemen</li> </ul>
<ul style="list-style-type: none"> <li>Vergleichen Sie die Funktionsweise eines Hubs mit der eines Switches.</li> <li>Worin besteht ein entscheidender Vorteil eines Switches?</li> </ul>	<ul style="list-style-type: none"> <li>Diskutieren Funktionalität, Leistungsfähigkeit und Zuverlässigkeit von Informatiksystemen</li> </ul>
<ul style="list-style-type: none"> <li>Erläutern Sie die Bedeutung von IP-Adresse und Subnetzmaske bei der Datenübertragung in einem TCP/IP-Netzwerk.</li> </ul>	<ul style="list-style-type: none"> <li>Wenden das Adressierungsprinzip (IP-Adresse, Subnetze) in Netzwerken auf der Basis des Internetprotokolls an</li> </ul>

**Die vier Computerfachräume dieser Schule (Raum 10 mit 23 Rechnern, Raum 11 mit 21 Rechnern, Raum 12 mit 18 Rechnern, Raum 13 mit 12 Rechnern) werden über einen File-Server miteinander verbunden und mittels Proxy-Server an das Internet angeschlossen.**

<ul style="list-style-type: none"> <li>• Beschreiben Sie eine entsprechende Netzwerktopologie.</li> <li>• Erstellen Sie eine statische IP-Adressierung für dieses Schulnetzwerk.</li> </ul>	<ul style="list-style-type: none"> <li>• Wenden das Adressierungsprinzip (IP-Adresse, Subnetze) in Netzwerken auf der Basis des Internetprotokolls an.</li> <li>• Erfassen, reflektieren und diskutieren informatische Sachverhalte aus nicht didaktisch aufbereiteten authentischen Texten.</li> </ul>
<ul style="list-style-type: none"> <li>• Das Verwaltungsnetz der Schule wird ebenfalls in das Netzwerk einbezogen. Kein Rechner aus den Räumen 10 bis 13 darf das Verwaltungsnetz erreichen.</li> <li>• Erläutern Sie eine Erweiterung des Netzwerkes um diese Forderung zu erfüllen.</li> </ul>	<ul style="list-style-type: none"> <li>• Beachten und reflektieren Aspekte der Datensicherheit bei der Kommunikation.</li> <li>• Beurteilen die Grenzen</li> </ul>
<ul style="list-style-type: none"> <li>• Der Schulleiter fordert eine Einsicht in die Log-Files (Protokoll aller Nutzeraktivitäten mit zugehöriger Uhrzeit) der Verwaltungsrechner.</li> <li>• Diskutieren Sie diese Forderung aus Sicht eines Verwaltungsangestellten.</li> </ul>	<ul style="list-style-type: none"> <li>• Beurteilen die Grenzen des Einsatzes von Informatiksystemen aufgrund individueller Verantwortung.</li> </ul>

**Eine Firma hat folgenden IP-Adressraum zugewiesen bekommen: 221.7.25.0/24. Sie benötigt mindestens 7 Subnetze.**

<ul style="list-style-type: none"> <li>• Entwickeln Sie ein Adressschema unter der Maßgabe, dass möglichst viele Hostadressen pro Subnetz zur Verfügung stehen.</li> <li>• Geben Sie die IP-Adresse des ersten nutzbaren Subnetzes, die Adresse des ersten nutzbaren Hosts und die Subnetzmaske an.</li> <li>• Wie viele nutzbare Hosts können pro Subnetz adressiert werden?</li> </ul>	<ul style="list-style-type: none"> <li>• Wenden das Adressierungsprinzip (IP-Adresse, Subnetze) in Netzwerken auf der Basis des Internetprotokolls an.</li> <li>• Erfassen, reflektieren und diskutieren informatische Sachverhalte aus nicht didaktisch aufbereiteten authentischen Texten.</li> </ul>
--	---

## 2 Softwareentwicklung (GK/LK)

	Eingangsvoraussetzungen Die Schülerinnen und Schüler	Abschlussorientierte Standards Die Schülerinnen und Schüler
<b>Informatisches Modellieren</b> <i>Modelle erstellen und bewerten</i>	<ul style="list-style-type: none"> <li>• beschreiben Modelle als vereinfachtes Abbild der realen Welt               <ul style="list-style-type: none"> <li>○ reale Sachverhalte werden abstrahiert, um sie dann mit einer künstlichen Mini-Umgebung (Schildkröte, Marienkäfer ...) bzw. einem virtuellen Robotermodell (Karol, Niki ...) abzubilden.</li> </ul> </li> <li>• hinterfragen und bewerten Ergebnisse einer Modellbildung kritisch               <ul style="list-style-type: none"> <li>○ Vergleich des virtuellen Arbeitsfeldes mit der Realität (z.B. Lagerhalle Karol vs. Lagerhalle echt)</li> <li>○ Herausarbeiten von zulässigen Vereinfachungen hinsichtlich der Wirklichkeit, aber auch eventueller Mängel des Modells</li> </ul> </li> <li>• wenden das Basiskonzept der objektorientierten Sichtweise auf Standardsoftware an               <ul style="list-style-type: none"> <li>○ kennen (intuitiv) die Begriffe Klasse, Objekt, Attribut, Methode,</li> <li>○ verstehen die Punktnotation der objektorientierten Schreibweise bei Standardsoftware.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• analysieren Realitätsausschnitte und wählen ein geeignetes Modellierungsverfahren aus,</li> <li>• variieren und erweitern vorgegebene Modelle,</li> <li>• entwickeln, implementieren, testen und validieren einfache Modelle,</li> <li>• reflektieren und beurteilen die eigene Modellierung,</li> <li>• beschreiben Basiskonzepte der objektorientierten Modellierung (Klasse, Objekt, Attribut, Methode),               <ul style="list-style-type: none"> <li>○ kennen grundlegende Begriffe der OOM und können sie an Beispielen richtig zuordnen</li> <li>○ wissen um die Bedeutung des Konstruktors</li> <li>○ kennen die Darstellung als UML-Diagramm und setzen die Modelle mit Hilfe einer Programmiersprache um</li> </ul> </li> <li>• entwerfen Methoden für die Manipulation von Objekten,               <ul style="list-style-type: none"> <li>○ Änderung der Attributwerte</li> </ul> </li> <li>• bilden Beziehungen zwischen Klassen ab,               <ul style="list-style-type: none"> <li>○ Hat-Beziehung</li> <li>○ Kennt-Beziehung</li> </ul> </li> </ul>
		nur LK: <ul style="list-style-type: none"> <li>• wenden rekursive Verfahren an,               <ul style="list-style-type: none"> <li>○ Analyse rekursiver Verfahren z. B. mithilfe der Schachtelmethode</li> <li>○ Implementierung mit einer imperativen Programmiersprache</li> <li>○ Rekursive Prädikate, Listenverarbeitung mit Prolog</li> </ul> </li> <li>• <i>Objektorientierte Modellierung</i>: wenden die Konzepte von Vererbung, Polymorphie und Kapselung an,               <ul style="list-style-type: none"> <li>○ Ist-Beziehung</li> </ul> </li> <li>• <i>Regelbasierte Modellierung</i>: wenden ein deklaratives Sprachparadigma zur Modellierung an,               <ul style="list-style-type: none"> <li>○ Grundlagen der Sprache Prolog</li> <li>○ Datenstrukturen in Prolog (Listen)</li> </ul> </li> </ul>

	<b>Eingangsvoraussetzungen Die Schülerinnen und Schüler</b>	<b>Abschlussorientierte Standards Die Schülerinnen und Schüler</b>
		<ul style="list-style-type: none"> <li>○ Übertragen Kenntnisse der Automatentheorie auf deren Umsetzung in einer deklarativen Programmiersprache</li> <li>○ Modellierung endlicher Automaten, Kellerautomaten und Turingmaschinen mit Prolog</li> <li>• unterscheiden Vor- und Nachteile der regelbasierten Modellierung.</li> </ul>
<p><b>Mit Information umgehen</b> <i>Information in Form von Daten darstellen und verarbeiten</i></p>	<ul style="list-style-type: none"> <li>• unterscheiden zwischen Information und Daten, <ul style="list-style-type: none"> <li>○ wissen, dass Informationen nur in Form von Daten mit Hilfe des Computers verarbeitet werden können.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• analysieren und strukturieren Informationen <ul style="list-style-type: none"> <li>○ abstrahieren Objekte aus der realen Welt</li> <li>○ können für gleichartige Objekte Klassen definieren,</li> <li>○ können aus der Problemstellung ableiten, welche Datenobjekte (Variablen, Konstanten) notwendig sind</li> <li>○ bei der Erstellung der Wissensbasis für ein Prologprogramm</li> </ul> </li> <li>• konstruieren Daten- bzw. Objektstrukturen und wenden auf diese geeignete Algorithmen bzw. Methoden an <ul style="list-style-type: none"> <li>○ wissen, dass Datenobjekte vereinbart (deklariert) werden müssen</li> <li>○ erkennen im Zusammenhang mit Variablen, was unter Initialisierung und Zuweisung zu verstehen ist</li> <li>○ können Datentypen der Problemsituation entsprechend einsetzen</li> <li>○ kennen unterschiedliche Standardalgorithmen (wie z.B. Such- und Sortierverfahren) und können diese implementieren</li> </ul> </li> <li>• unterscheiden zwischen Syntax und Semantik und erläutern dies in natürlichen, halb formalen und formalen Sprachen <ul style="list-style-type: none"> <li>○ im Zusammenhang mit der verwendeten Programmiersprache wird die syntaktische Analyse als eine Aufgabe des Compilers bzw. Interpreters thematisiert</li> </ul> </li> </ul> <p>nur LK:</p> <ul style="list-style-type: none"> <li>• implementieren zusammengesetzte und dynamische Daten- bzw. Objektstrukturen (Listen, Bäume) und wenden diese an.</li> </ul>

	<b>Eingangsvoraussetzungen</b> <b>Die Schülerinnen und Schüler</b>	<b>Abschlussorientierte Standards</b> <b>Die Schülerinnen und Schüler</b>
<b>Informatiksysteme verstehen</b> <i>Wirkprinzipien kennen und anwenden</i>	<ul style="list-style-type: none"> <li>• erläutern Eigenschaften von Algorithmen an einfachen Beispielen               <ul style="list-style-type: none"> <li>○ formulieren Beispiele für Alltagsalgorithmen, Algorithmen aus der Mathematik und aus der Arbeit mit der gewählten Mini-Umgebung bzw. dem virtuellen Robotermodell und begründen an Hand der Eigenschaften, dass es sich um algorithmische Abläufe handelt</li> <li>○ kennen alle algorithmischen Grundstrukturen (Sequenz, Auswahl, Wiederholung) und können diese zielgerichtet anwenden.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• vergleichen formale und natürliche Sprachen,</li> <li>• diskutieren Funktionalität, Leistungsfähigkeit und Zuverlässigkeit von Informatiksystemen               <ul style="list-style-type: none"> <li>○ an Hand ihrer erstellten Softwareprodukte, (z. B.: „Arbeitet das Programm <u>bei allen</u> möglichen Eingaben immer korrekt?“)</li> </ul> </li> <li>• beurteilen Algorithmen hinsichtlich ihrer Effizienz,               <ul style="list-style-type: none"> <li>○ z. B. Aufwandsanalyse bei Sortierverfahren</li> </ul> </li> </ul> <p>nur LK:</p> <ul style="list-style-type: none"> <li>• konstruieren Software unter Beachtung des Prinzips der Modularisierung (Schnittstellen),</li> <li>• ordnen Algorithmen gegebenen Komplexitätsklassen zu               <ul style="list-style-type: none"> <li>○ z. B. „Handlungsreisender“ als schwer lösbar,</li> <li>○ „Halteproblem“ als nicht lösbares Problem.</li> </ul> </li> <li>• analysieren und konstruieren formale Sprachen und beschreiben beispielhaft den Zusammenhang zwischen Automaten und Grammatiken               <ul style="list-style-type: none"> <li>○ Syntaxanalyse mit Prolog</li> </ul> </li> </ul>

<p><b>Problemlösen</b> <i>Probleme erfassen und mit Informatiksystemen lösen</i></p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• beschreiben algorithmische Abläufe umgangssprachlich und grafisch, <ul style="list-style-type: none"> <li>○ sind in der Lage Algorithmen in verschiedenen Darstellungsformen zu beschreiben</li> </ul> </li> <li>• modellieren einfache Abläufe mit Algorithmen (Sequenz, Auswahl, Wiederholung)</li> <li>• setzen Algorithmen in Programme um <ul style="list-style-type: none"> <li>○ kennen und nutzen (selbstdefinierte) Anweisungen (Methoden, Prozeduren, Funktionen) und Bedingungen (Sensoren) in einfachen Programmierumgebungen (Mini-Welten).</li> </ul> </li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• wenden die Phasen des Problemlöseprozesses (informelle Problembeschreibung, formale Modellierung, Implementierung und Realisierung, Bewertung und Modellkritik) an,</li> <li>• setzen informatische Methoden (z. B. Bottom-Up, Top-Down, Modularisierung, Prototyping) zielorientiert ein,</li> <li>• setzen im Problemlöseprozess einfache Entwicklungswerkzeuge ein</li> </ul>
<p><b>Kommunizieren und Kooperieren</b> <i>Teamarbeit organisieren und koordinieren</i></p>	<ul style="list-style-type: none"> <li>• verwenden im angemessenen Rahmen die Fachsprache</li> <li>• dokumentieren und präsentieren ihre Arbeitsergebnisse</li> </ul>	<p>nur LK:</p> <ul style="list-style-type: none"> <li>• wählen informatische Werkzeuge zur Problemlösung selbstständig aus und begründen die getroffene Auswahl (z. B. Programmiersprachenparadigma),</li> <li>• zeigen Grenzen des Problemlösens mit Informatiksystemen auf</li> <li>• Graphen (kürzester Weg)</li> <li>• „Halteproblem“</li> </ul> <ul style="list-style-type: none"> <li>• verfügen über eine angemessene Fachsprache und verwenden sie sachgerecht, <ul style="list-style-type: none"> <li>○ Algorithmus, Daten, Objekt, Klasse, Attribut ...</li> <li>○ Fakten, Regeln, Prädikate, Anfragen, Listen</li> </ul> </li> <li>• verwenden selbstständig Fachtexte, Dokumentationen und Hilfesysteme, <ul style="list-style-type: none"> <li>○ nutzen zielgerichtet die Hilfe des Entwicklungswerkzeugs</li> </ul> </li> <li>• dokumentieren, visualisieren, präsentieren und verteidigen Ergebnisse der Teamarbeit <ul style="list-style-type: none"> <li>○ im Rahmen eines Softwareprojekts.</li> </ul> </li> </ul> <p>nur LK:</p> <ul style="list-style-type: none"> <li>• planen, organisieren und leiten selbstständig Projektarbeit.</li> </ul>

## Zeitlicher Verlauf und methodische Hinweise

### Kurshalbjahr 2: Grundlegende Datenstrukturen (Automaten)

(parallel zum Thema Sprache und

Schwerpunkte, Zeit	fachliche Inhalte/ Begriffe	methodisch-didaktische Hinweise	Beispiele, Kontext
Algorithmische Grundstrukturen	<ul style="list-style-type: none"> <li>Wiederholung von algorithmischen Grundstrukturen (Steuerstrukturen);</li> <li>Ergänzung bezüglich der Eingangsvoraussetzungen, z. B. mehrseitige Verzweigung, Zählschleife</li> <li>Variablenkonzept,</li> <li>Einfache Datentypen,</li> </ul>	<ul style="list-style-type: none"> <li>Umgang mit Editor, vorgegebene Oberfläche</li> <li>Verschiedene Darstellungsformen von Algorithmen (als Struktogramm bzw. Quelltext der Programmiersprache)</li> <li>Das Variablenkonzept als zentraler neuer Aspekt (vorher: parameterfreie Betrachtung)</li> <li>Schüler setzen Typumwandlungen gezielt ein</li> </ul>	<ul style="list-style-type: none"> <li>Einfache Problemstellungen wie z. B. Verzinsung von Kapital</li> <li>Fachübergreifender Aspekt zu den Fächern Deutsch, Englisch: Syntax und Semantik</li> <li>Natürliche und künstliche Sprachen.</li> <li>Variablenbegriff aus der Mathematik.</li> <li>Unterschiede herausstellen, z. B. <math>x \leftarrow x+1</math></li> <li>Datentypen im weitesten Sinne schon bei Standardsoftware prinzipiell thematisiert</li> </ul>
Unterprogrammtechnik	<ul style="list-style-type: none"> <li>Modularisierung</li> <li>Parameterkonzept</li> <li>MVC-Konzept</li> </ul>	<ul style="list-style-type: none"> <li>Gültigkeitsbereich von Variablen (lokal/global).</li> <li>Das Parameterkonzept als Grundlage für den Datenaustausch zwischen Teilen (Modulen) eines Programms.</li> <li>Die Schülerinnen und Schüler kennen die unterschiedlichen Parameterarten und können diese gezielt einsetzen.</li> </ul>	<ul style="list-style-type: none"> <li>Analogie zur Modularisierung in anderen Bereichen:</li> <li>HiFi-Anlagen sind modular aufgebaut</li> <li>Einzelne Produktionsabschnitte bei der Herstellung von Produkten wie z. B. Autos etc.</li> </ul>
Erweiterte Datenstrukturen	<ul style="list-style-type: none"> <li>Datenstruktur Zeichenkette (String)</li> </ul>	<ul style="list-style-type: none"> <li>Bisherige, überwiegend numerisch orientierte Anwendungsbeispiele in Richtung auf eine allgemeine Manipulation von Texten erweitern.</li> </ul>	<ul style="list-style-type: none"> <li>Alltagserfahrung der Schüler: Name eines Schülers, Seriennummer, ISBN eines Lehrbuches, EAN ...</li> <li>Vorschlag: Einführung in die Kryptologie; einfache Verfahren wie z. B. Verschlüsselung nach Caesar</li> <li>Wechselwirkung Informatik – Gesellschaft!</li> <li>(sichere Kommunikation im Internet, Verschlüsselung wichtiger Dokumente auf der Festplatte, biometrischer Reisepass usw.)</li> </ul>

Schwerpunkte, Zeit	fachliche Inhalte/ Begriffe	methodisch-didaktische Hinweise	Beispiele, Kontext
Erweiterte Datenstrukturen	<ul style="list-style-type: none"> <li>Datenstruktur Array</li> </ul>	<ul style="list-style-type: none"> <li>Häufigkeitsanalyse eines Caesar-verschlüsselten Textes als Motivation zur Einführung der Datenstruktur Array</li> <li>Verwendung unterschiedlicher Datentypen ermöglicht unterschiedliche Lösungsstrategien.</li> <li>Vergleich: Erfassen der Zahlen als Array of Boolean, Array of Integer, Menge</li> </ul>	<ul style="list-style-type: none"> <li>Buchstabenhäufigkeit in einem deutschen Text.</li> <li>(im Vergleich z. B. zu einem englischen)</li> </ul> <p>Weiteres Beispiel:</p> <ul style="list-style-type: none"> <li>Simulation der Ziehung der Lottozahlen</li> </ul>

### Kurshalbjahr 3: Modellentwicklung und Implementierung (*alleiniges Thema*)

Schwerpunkte, Zeit	fachliche Inhalte/ Begriffe	methodisch-didaktische Hinweise	Beispiele, Kontext
Grundbegriffe und Grundlagen der OOM	<ul style="list-style-type: none"> <li>Systematisierung und Erweiterung der Begriffe Klasse, Objekt, Exemplar, Zustand, Klassenattribute, Klassenmethoden, UML</li> </ul>	<ul style="list-style-type: none"> <li>Modell als Abbild der Realität.</li> <li>MVC-Konzept wieder aufgreifen.</li> <li>Die Schülerinnen und Schüler sind in der Lage, gleichartige Objekte zu Klassen zu abstrahieren.</li> <li>Verwendung bzw. Nutzung vorgegebener Klassen und ihrer Methoden.</li> <li>Die Schülerinnen und Schüler können eine Klassendefinition in UML-Schreibweise darstellen.</li> <li>Vergleich UML-Diagramm mit Notation in der benutzten Programmiersprache</li> </ul>	<ul style="list-style-type: none"> <li>Einführung am Beispiel geometrischer Figuren (z. B. Kreis, Rechteck)</li> </ul>

Schwerpunkte, Zeit	fachliche Inhalte/ Begriffe	methodisch-didaktische Hinweise	Beispiele, Kontext
Modellierung und Implementierung	<ul style="list-style-type: none"> <li>Aufbau von Klassen, Kapselung,</li> <li>Konstruktor</li> </ul>	<ul style="list-style-type: none"> <li>Analyse und Erweiterung vorgefertigter Klassen.</li> <li>Methoden zur Manipulation von Objekten entwickeln.</li> <li>Die Schülerinnen und Schüler kennen die zur Implementierung erforderlichen Elemente der Programmiersprache und können Sie sachgerecht anwenden.</li> </ul>	<ul style="list-style-type: none"> <li>Klassen Kreis, Rechteck um neue Attribute und Methoden erweitern. (z.B. Fläche oder Umfang und deren Berechnung)</li> </ul>
	<ul style="list-style-type: none"> <li>Beziehungen zwischen Klassen</li> </ul>	<ul style="list-style-type: none"> <li>Kennt-Beziehung</li> <li>Hat-Beziehung</li> <li>Ist-Beziehung</li> </ul>	<ul style="list-style-type: none"> <li>Klassenhierarchie am Beispiel geometrischer Figuren</li> </ul>
	<ul style="list-style-type: none"> <li>Beziehungen zwischen Klassen</li> </ul>	<ul style="list-style-type: none"> <li>Beispiel für Zusammenspiel mehrerer Klassen</li> <li>(eventuell Verbindung zu vorherigem: Klasse Ampel nutzt Klasse Kreis)</li> </ul>	<ul style="list-style-type: none"> <li>Ampel</li> <li>Ableiten der Klassen „Kfz-Ampel“ und „Fußgänger-Ampel“ von der Oberklasse „Ampel“</li> </ul>
	<p>Beziehungen zwischen Klassen. Schrittfolge:</p> <ol style="list-style-type: none"> <li>Es existiert nur eine Klasse (tKonto)</li> <li>Hinzu kommt eine Klasse tBank und als Beziehung eine Hat-Beziehung</li> <li>Klasse tKunde</li> <li>Kennt-Beziehung zwischen tKonto und tKunde</li> <li>Vererbung (Ist-Beziehung) von allgemeiner Klasse tKonto zu tGirokonto und tSparkonto.</li> </ol>	<ul style="list-style-type: none"> <li>Erste komplexere Strukturen losgelöst vom geometrischen Hintergrund</li> <li>Schrittweise Umfang und Schwierigkeitsgrad erhöhen.</li> </ul>	<ul style="list-style-type: none"> <li>Konto-Bank</li> <li>(Verwaltung von Konten in einer Bank)</li> </ul>
	<ul style="list-style-type: none"> <li>Standardverfahren wie Suchen und Sortieren</li> </ul>	<ul style="list-style-type: none"> <li>Selbstständige Schülertätigkeit.</li> </ul>	<ul style="list-style-type: none"> <li>Zum Beispiel:</li> <li>Verwaltung von Schülern eines Jahrgangs</li> <li>Verwaltung der Kunden und der Autos im Autohaus</li> </ul>

**Kurshalbjahr 3: Modellentwicklung und Implementierung**  
(Deklarative Programmiersprache Prolog)

<b>Schwerpunkte, Zeit</b>	<b>fachliche Inhalte/ Begriffe</b>	<b>methodisch-didaktische Hinweise</b>	<b>Beispiele, Kontext</b>
Grundbegriffe und Grundlagen eines Prolog-Systems	<ul style="list-style-type: none"> <li>• Fakten, Regeln, Anfragen</li> <li>• Konstanten, Variablen</li> <li>• Prädikate, Rekursive Prädikate</li> <li>• Anonyme Variable</li> </ul>	<ul style="list-style-type: none"> <li>• An Hand einfacher Beispiele die Begrifflichkeiten erklären.</li> <li>• UND- bzw. ODER-Anfrage</li> </ul>	<ul style="list-style-type: none"> <li>• Stammbaum</li> </ul>
Einfache Datenbankkonzepte	<ul style="list-style-type: none"> <li>• Selektion, Projektion</li> </ul>	<ul style="list-style-type: none"> <li>• Herausarbeiten des Unterschieds: Auswahl bestimmter Datensätze zu Auswahl bestimmter Datenfelder</li> </ul>	<ul style="list-style-type: none"> <li>• Einfache Auskunftssysteme (Fahrplan, Hotelangebote)</li> </ul>
Datenstruktur Liste	<ul style="list-style-type: none"> <li>• Listenoperationen</li> <li>• Kopf-Rest-Methode</li> </ul>	<ul style="list-style-type: none"> <li>• Rekursion bei der Verarbeitung von Listen</li> <li>• Baumstruktur bei Listen</li> </ul>	<ul style="list-style-type: none"> <li>• Labyrinth</li> </ul>
Sprache und Automaten	<ul style="list-style-type: none"> <li>• Modellierung von Grammatiken und Automaten in Prolog</li> </ul>	<ul style="list-style-type: none"> <li>• viele Inhalte des Themenfeldes Sprachen und Automaten können auch an dieser Stelle thematisiert werden</li> </ul>	<ul style="list-style-type: none"> <li>• Syntaxdiagramme</li> <li>• Charakterisierung unterschiedlicher Automatenmodelle</li> </ul>

## Testaufgaben

### Würfelspiel (Anregung bzw. Quelle: Dümmler, Klausur-Aufgaben Informatik, A 21.13)

#### Problembeschreibung:

Ein Spieler spielt gegen die Bank. Er wirft wiederholt einen Würfel. Er hat das Spiel verloren, sobald er eine gerade Zahl gewürfelt hat. Er hat gewonnen, sobald eine 1 und eine 3 (in beliebiger Reihenfolge und nicht zwingend aufeinanderfolgend) gefallen sind. Wenn er eine 5 würfelt, wird das Spiel fortgesetzt und er darf weiterwürfeln.

Beispiele für Gewinnfolgen:

1, 5, 3  
5, 5, 3, 3, 5, 1

Beispiele für Verlustfolgen:

1, 5, 2  
4  
3, 5, 6

## Aufgaben

## Abschlussorientierter Standard

Aufgabe 1: Die beschriebenen Spielregeln lassen sich mit Hilfe eines endlichen Automaten (Akzeptor) A modellieren.

a) Geben Sie die Definition eines möglichen Automaten A an. Stellen Sie die Überföhrungsfunktion des Akzeptors mit Hilfe eines Zustandsgraphen dar.

Verfügen über eine angemessene Fachsprache und verwenden sie sachgerecht. Modellieren automatisierte Abläufe mithilfe endlicher Automaten.

b) Beschreiben Sie welche Bedeutung ihren gewählten Zuständen zukommt.

Erläutern Basiskonzepte der zustandsorientierten Modellierung. Verfügen über eine angemessene Fachsprache und verwenden sie sachgerecht.

Aufgabe 2: Der Automat lässt sich unter objektorientierter Sichtweise als Klasse in Form eines UML-Klassendiagramms erfassen.

a) Entwerfen Sie ein Klassendiagramm zur Modellierung des Automaten.

Analysieren Realitätsausschnitte. Beschreiben Basiskonzepte der objektorientierten Modellierung.

b) Implementieren Sie die notwendigen Methoden, um das Spiel nach obigen Regeln zu simulieren.

Entwerfen Methoden für die Manipulation von Objekten. Setzen informatische Methoden zielorientiert ein.

c) Erstellen Sie eine allgemeine Basis-klass für eine Klasse CSpiel, aus der sich ihre in der Aufgabe 2a) entworfene Klasse durch Vererbung ableiten lässt. Verändern Sie ihr UML-Diagramm aus 2a) dementsprechend. Erläutern Sie daran die Begriffe Vererbung und Polymorphie.

Beschreiben Basiskonzepte der objektorientierten Modellierung. Bilden Beziehungen zwischen Klassen ab. Wenden die Konzepte von Vererbung, Polymorphie und Kapselung an. Verfügen über eine angemessene Fachsprache und verwenden sie sachgerecht.

## Aufgaben

## Abschlussorientierter Standard

Die Spielregeln werden wie folgt geändert: Zusätzlich zu den oben beschriebenen Gewinnmöglichkeiten kommt jetzt hinzu, dass der Spieler auch siegt, wenn zwei Dreien direkt hintereinander fallen.

d) Diskutieren Sie ihre Modellierung aus den Teilaufgaben 2a und 2b unter diesem neuen Gesichtspunkt.	Reflektieren und beurteilen die eigene Modellierung.
e) Passen Sie den in Aufgabe 1 erstellten Zustandsgraph der neuen Situation entsprechend an.	Analysieren Realitätsausschnitte. Variieren und erweitern vorgegebene Modelle.

Alternativaufgabe 2a - c Der Automat lässt sich mit Hilfe eines Prolog-Programms umsetzen

a) Modellieren Sie den Automaten in Prolog durch Angabe der Prädikate für Eingabe, Zustände und Zustandsübergänge.	Analysieren Realitätsausschnitte.
b) Erstellen Sie die notwendigen Regeln für den Automaten unter Verwendung von Rekursion und des Cut. Erläutern Sie beispielhaft die Idee der Rekursion und die Wirkungsweise des Cut.	Beschreiben Basiskonzepte der logischen Programmierung. Entwerfen Methoden für die Manipulation von Objekten. Setzen informatische Methoden zielorientiert ein.
c) Formulieren Sie je zwei Anfragen für Gewinn- bzw. Verlustsituationen.	

## Erwartungshorizont Grundkurs

Automat

1a) Definition  $M = (\{s_0, s_1, s_2, sv, sg\}, \{1, 2, 3, 4, 5, 6\}, \delta, s_0, \{sv, sg\})$

Übergangsgraph

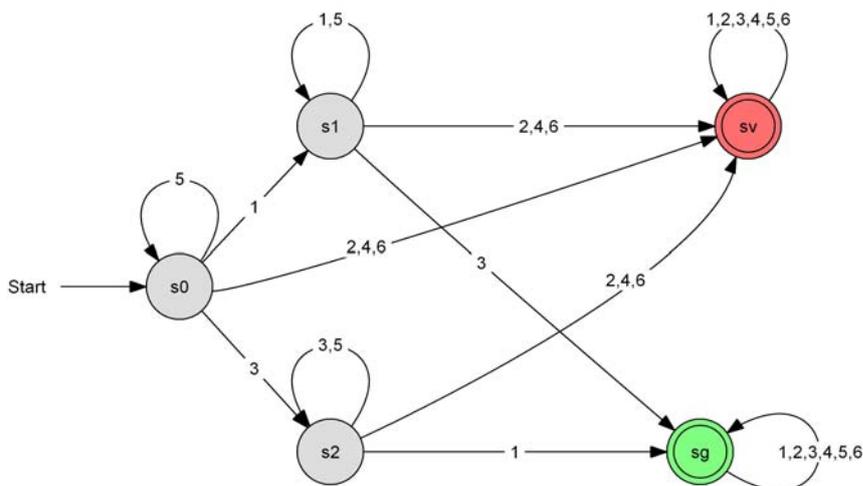


Tabelle:

	1	2	3	4	5	6
s0	s1	sv	s2	sv	s0	sv
s1	s1	sv	sg	sv	s1	sv
s2	sg	sv	s2	sv	s2	sv
sv						
sg						

Zu 1b)

- s<sub>0</sub>: Startzustand. In diesem Zustand verbleibt der Akzeptor, solange ausschließlich Fünfen gewürfelt werden.
- s<sub>1</sub>: Dieser Zustand ist dadurch gekennzeichnet, dass ausschließlich Fünfen oder Einsen gewürfelt wurden.
- s<sub>2</sub>: Dieser Zustand ist dadurch gekennzeichnet, dass ausschließlich Fünfen oder Dreien gewürfelt wurden.
- s<sub>v</sub>: „Verlust-Zustand“. In diesem Zustand gelangt (und verbleibt) A, wenn mindestens eine gerade Zahl gewürfelt wurde.
- s<sub>G</sub>: „Gewinn-Zustand“. A gelangt (und verbleibt) in ihm, wenn mindestens eine Eins und mindestens eine Drei und sonst nur Fünfen gewürfelt werden.

2)

z<sub>g</sub> bedeutet Zustand gewonnen, z<sub>v</sub> verloren

		Gewürfelte Zahl					
		1	2	3	4	5	6
Zustände	z <sub>0</sub>	z <sub>1</sub>	z <sub>v</sub>	z <sub>2</sub>	z <sub>v</sub>	z <sub>0</sub>	z <sub>v</sub>
	z <sub>1</sub>	z <sub>1</sub>	z <sub>v</sub>	z <sub>g</sub>	z <sub>v</sub>	z <sub>1</sub>	z <sub>v</sub>
	z <sub>2</sub>	z <sub>g</sub>	z <sub>v</sub>	z <sub>2</sub>	z <sub>v</sub>	z <sub>2</sub>	z <sub>v</sub>
	z <sub>g</sub>	z <sub>g</sub>	z <sub>g</sub>	z <sub>g</sub>	z <sub>g</sub>	z <sub>g</sub>	z <sub>g</sub>
	z <sub>v</sub>	z <sub>v</sub>	z <sub>v</sub>	z <sub>v</sub>	z <sub>v</sub>	z <sub>v</sub>	z <sub>v</sub>

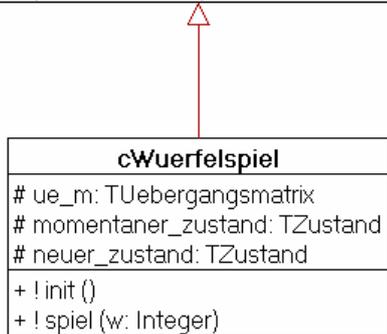
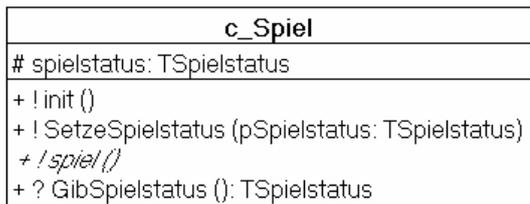
Tzustand = (z<sub>0</sub>,z<sub>1</sub>,z<sub>2</sub>,z<sub>g</sub>,z<sub>v</sub>);

Tuebergangsmatrix = Array[1..6,z<sub>0</sub>..z<sub>2</sub>] of TZustand;

<b>cWurfelspiel</b>	
#spielstatus:	TSpielstatus
#ue_m:	TUebergangsmatrix
#momentaner_zustand:	TZustand
#neuer_zustand:	TZustand
+!create ()	
+!SetzeSpielstatus (pSpielstatus: TSpielstatus)	
+!spiel (w: Integer)	
+?GibSpielstatus (): TSpielstatus	

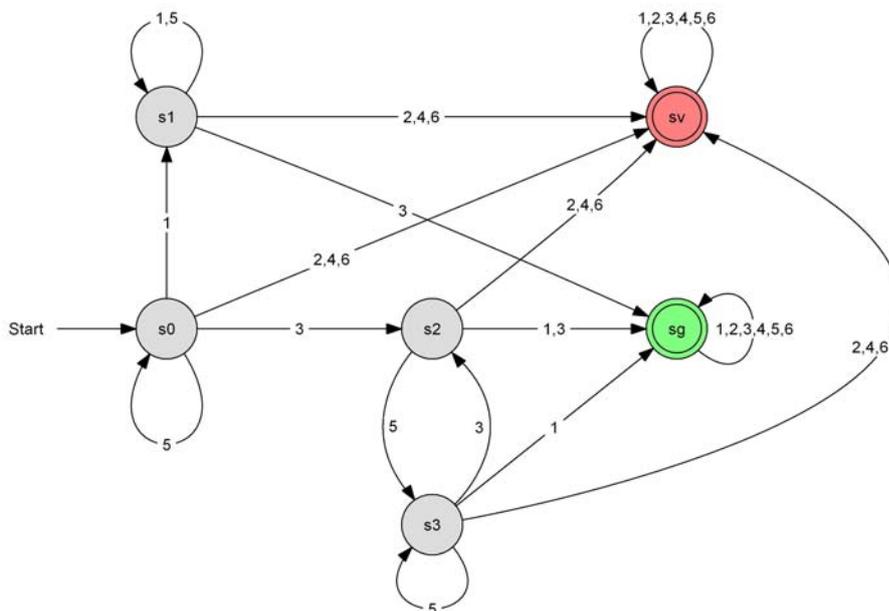
## Erwartungshorizont Leistungskurs

2) a, c



2) e Definition  $M = (\{s_0, s_1, s_2, sv, sg, s_3\}, \{1, 2, 3, 4, 5, 6\}, \delta, s_0, \{sv, sg\})$

Übergangsgraph



Tabelle

	1	2	3	4	5	6
s0	s1	sv	s2	sv	s0	sv
s1	s1	sv	sg	sv	s1	sv
s2	sg	sv	sg	sv	s3	sv
sv						
sg						
s3	sg	sv	s2	sv	s3	sv

### 3 Sprachen und Automaten (LK)

	<b>Eingangsvoraussetzungen</b> Die Schülerinnen und Schüler	<b>Abschlussorientierte Standards</b> Die Schülerinnen und Schüler
<b>Informatisches Modellieren</b>  <i>Modelle erstellen und bewerten</i>	<ul style="list-style-type: none"> <li>• beschreiben Modelle als vereinfachtes Abbild der realen Welt,               <ul style="list-style-type: none"> <li>○ kennen das Konzept der OOM bei Standardanwendungen</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• analysieren Realitätsausschnitte und wählen ein geeignetes Modellierungsverfahren aus,               <ul style="list-style-type: none"> <li>○ modellieren mithilfe des Modells Endlicher Automat,</li> </ul> </li> <li>• variieren und erweitern vorgegebene Modelle,               <ul style="list-style-type: none"> <li>○ entwickeln, implementieren mit einer höheren Programmiersprache oder mit geeigneten Tools (z. B. Automaten-Kara),</li> <li>○ testen und validieren einfache Modelle,</li> </ul> </li> <li>• reflektieren und beurteilen die eigene Modellierung,               <ul style="list-style-type: none"> <li>○ überprüfen an Hand von Ablaufprotokollen bzw. Simulationsprogrammen die korrekte Arbeitsweise der Automatenmodelle</li> </ul> </li> <li>• erläutern Basiskonzepte der zustandsorientierten Modellierung,               <ul style="list-style-type: none"> <li>○ kennen die mathematische Definition des Automaten und stellen die Überföhrungsfunktion als Zustandsgraph dar</li> </ul> </li> <li>• modellieren automatisierte Abläufe mit Hilfe endlicher Automaten               <ul style="list-style-type: none"> <li>○ zur Analyse formaler Sprachen. (z. B. Akzeptoren )</li> <li>○ zur Abbildung endlicher Automaten mit Ausgabe (z. B. Transduktoren)</li> </ul> </li> </ul>
<b>Mit Information umgehen</b>  <i>Information in Form von Daten darstellen und verarbeiten</i>		<ul style="list-style-type: none"> <li>• unterscheiden zwischen Syntax und Semantik und erläutern dies in natürlichen, halbformalen und formalen Sprachen.               <ul style="list-style-type: none"> <li>○ erläutern die Begriffe Syntax und Semantik und arbeiten diesbezüglich Gemeinsamkeiten und Unterschiede von natürlichen, halbformalen und formalen Sprachen heraus</li> </ul> </li> </ul>

<p><b>Informatiksysteme verstehen</b></p> <p><i>Wirkprinzipien kennen und anwenden</i></p>	<ul style="list-style-type: none"> <li>• erläutern Eigenschaften von Algorithmen an einfachen Beispielen, <ul style="list-style-type: none"> <li>○ kennen Eigenschaften von Algorithmen aus der Arbeit mit Standardanwendungen und verschiedenen Programmierwerkzeugen und aus dem täglichen Leben</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• beschreiben den Computer als programmierbaren, universellen Automaten, <ul style="list-style-type: none"> <li>○ stellen Analogiebetrachtung zwischen Computer und endlichen Automaten an</li> <li>○ erkennen, dass eine vollständige Beschreibung des Computers mit Hilfe des Modells endlicher Automaten möglich ist.</li> </ul> </li> <li>• vergleichen formale und natürliche Sprachen, <ul style="list-style-type: none"> <li>○ kennen Gemeinsamkeiten (Alphabet, Syntax, Semantik) und Unterschiede (Sprachgefüge, zeitliche Veränderlichkeit von Vokabular und Grammatik) der einzelnen Sprachen</li> </ul> </li> <li>• ordnen Algorithmen gegebenen Komplexitätsklassen zu, <ul style="list-style-type: none"> <li>○ kennen die Grenzen der Berechenbarkeit (TURING-Berechenbarkeit) und erkennen, dass nicht alles, was prinzipiell berechenbar ist, sich auch praktisch programmieren lässt</li> <li>○ ausgehend von den zur Verfügung stehenden Mitteln erfolgt eine Einteilung der Probleme in Komplexitätsklassen</li> </ul> </li> <li>• analysieren den Aufbau und die Arbeitsweise eines allgemeinen Maschinenmodells (Turing-Maschine oder Registermaschine), <ul style="list-style-type: none"> <li>○ Einsatz zur Sprachanalyse und</li> <li>○ bei Problemen der Berechenbarkeit</li> </ul> </li> <li>• analysieren und konstruieren formale Sprachen und beschreiben beispielhaft den Zusammenhang zwischen Automaten und Grammatiken. <ul style="list-style-type: none"> <li>○ ordnen den Sprachklassen die entsprechenden Automatenmodelle zu</li> </ul> </li> </ul>
--	---	--

<p><b>Problemlösen</b></p> <p><i>Probleme erfassen und mit Informatiksystemen lösen</i></p>	<ul style="list-style-type: none"> <li>• beschreiben algorithmische Abläufe umgangssprachlich und grafisch, <ul style="list-style-type: none"> <li>○ formulieren Algorithmen verbal und arbeiten mit Struktogrammen und/oder PAP</li> </ul> </li> <li>• modellieren einfache Abläufe mit Algorithmen (Sequenz, Auswahl, Wiederholung), <ul style="list-style-type: none"> <li>○ sind in der Lage Vorgänge aus dem täglichen Leben bzw. mathematische Berechnungen algorithmisch zu modellieren</li> </ul> </li> <li>• setzen Algorithmen in Programme um. <ul style="list-style-type: none"> <li>○ arbeiten selbstständig mit ausgewählten Werkzeugen</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• wenden die Phasen des Problemlöseprozesses an: informelle Problembeschreibung, formale Modellierung, Implementierung und Realisierung, Bewertung und Modellkritik <ul style="list-style-type: none"> <li>○ an Problemen der lexikografischen Analyse des Compilers (Parser) erfolgt die Modellierung mit Endlichen Automaten, die Simulation mit Programmen und das kritische Hinterfragen der Ergebnisse der Modellierung</li> </ul> </li> <li>• setzen informatische Methoden (z. B. Bottom-Up, Top-Down) bei der Syntaxanalyse ein <ul style="list-style-type: none"> <li>○ an kleinen Ausschnitten von natürlichen oder Programmiersprachen erfolgt die syntaktische Analyse</li> </ul> </li> <li>• zeigen Grenzen des Problemlösens mit Informatiksystemen am Beispiel auf. <ul style="list-style-type: none"> <li>○ erläutern die Grenzen des Problemlösens an selbst gewählten Beispielen</li> </ul> </li> </ul>
<p><b>Kommunizieren und kooperieren</b></p> <p><i>Teamarbeit organisieren und koordinieren</i></p>	<ul style="list-style-type: none"> <li>• verwenden im angemessenen Rahmen die Fachsprache,</li> <li>• dokumentieren und präsentieren ihre Arbeitsergebnisse.</li> </ul>	<ul style="list-style-type: none"> <li>• verfügen über eine angemessene Fachsprache und verwenden sie sachgerecht, <ul style="list-style-type: none"> <li>○ verwenden und kennen die Begriffe Alphabet, Wort, Wortmenge, Sprache, Syntax, Semantik, Grammatik, Endlicher Automat, TURING-Maschine ...</li> </ul> </li> <li>• verwenden selbstständig Fachtexte, Dokumentationen und Hilfesysteme, <ul style="list-style-type: none"> <li>○ bei der Erarbeitung des Vergleichs Endlicher Automat und Computer bzw. bei Problemen zur Berechenbarkeitstheorie bietet sich das selbständige Arbeiten mit Fachtexten an</li> </ul> </li> </ul>
<p><b>Wechselwirkung zwischen Informatiksystemen, Mensch und Gesellschaft beurteilen</b></p> <p><i>Anwendungen erfassen und Auswirkungen abschätzen</i></p>		<ul style="list-style-type: none"> <li>• bewerten Risiken und Chancen von Informatiksystemen <ul style="list-style-type: none"> <li>○ Auswirkungen des Einsatzes von Automaten auf die Arbeitswelt</li> </ul> </li> </ul>

## Zeitlicher Verlauf und methodische Hinweise

Schwerpunkte der UE (Zeit)	Inhalte/Begriffe	methodisch/didaktische Hinweise	Kontext (Erfahrungswelt der Schüler, weitere Themen, andere Fächer)
Klassifikation von Sprachen	<ul style="list-style-type: none"> <li>• Unterschiede und Gemeinsamkeiten natürlicher und künstlicher Sprachen</li> <li>• Grammatik, Syntax, Semantik, Pragmatik</li> </ul>	<ul style="list-style-type: none"> <li>• Schüler erarbeiten die Definitionen der Begriffe unter Einbeziehung ihres Wissens aus dem Deutsch- und Fremdsprachenunterricht und erläutern diese an Beispielen</li> </ul>	<ul style="list-style-type: none"> <li>• Schüler kennen natürliche Sprachen (Deutsch, Englisch, Französisch...) Programmiersprachen und die mathematische Formelsprache</li> </ul>
Beschreibungen formaler Sprachen	<ul style="list-style-type: none"> <li>• Sprachen für die Mensch-Maschine-Kommunikation müssen für die automatische Verarbeitung geeignet sein</li> <li>• Syntaxdiagramme, (EBNF)</li> <li>• Bottom up- und Topdown-Ableitung</li> <li>• Ableitungsbaum</li> </ul>	<ul style="list-style-type: none"> <li>• Ausgangspunkt: Wie muss eine Sprache geschaffen sein, damit sie für die Mensch-Maschine-Kommunikation geeignet ist?</li> <li>• Übertragung dieser Erkenntnisse auf Ausschnitte von natürlichen Sprachen und Überleitung zu einer bekannten höheren Programmiersprache</li> <li>• Darstellung der Sprachregel mithilfe von Syntaxdiagrammen</li> <li>• Schüler weisen durch Ableitungen nach, ob ein zur Sprache gehörendes Wort (Satz) vorliegt.</li> </ul>	<ul style="list-style-type: none"> <li>• Schüler stellen immer einen Bezug zur deutschen Sprache (bzw. Fremdsprache) her</li> </ul>
Definition einer Grammatik	<ul style="list-style-type: none"> <li>• Grammatik als 4-Tupel</li> <li>• Terminal- und Nichtterminalsymbole, Produktionen (Regeln), Startsymbol</li> </ul>	<ul style="list-style-type: none"> <li>• Schüler formalisieren die bisherigen Überlegungen → Definition der Grammatik</li> </ul>	<ul style="list-style-type: none"> <li>• Bezug zu Deutsch und Fremdsprachen</li> </ul>
Klassifikation von Grammatiken Linkslineare Sprachen	<ul style="list-style-type: none"> <li>• Wortproblem und Syntaxanalyseproblem müssen effizient lösbar sein</li> <li>• Aufbau der Regeln einer linkslinearen (rechtslinearen) Grammatik</li> </ul>	<ul style="list-style-type: none"> <li>• Schüler erkennen, dass viele Fragen offen bleiben, wenn alle möglichen Regeln zugelassen werden. → strenge Forderungen an den Aufbau der Regeln notwendig</li> </ul>	<ul style="list-style-type: none"> <li>• Aufgaben eines Compilers: Erkennung der einzelnen Anweisungen und Übertragung in Maschinsprache</li> <li>• Scanner: lexikalische Analyse (Zerlegung in Zeichen/Token)</li> <li>• Parser: Syntaxanalyse (Erzeugung eines Ableitungsbaumes)</li> </ul>

Schwerpunkte der UE (Zeit)	Inhalte/Begriffe	methodisch/didaktische Hinweise	Kontext (Erfahrungswelt der Schüler, weitere Themen, andere Fächer)
Endliche Automaten (Akzeptor)	<ul style="list-style-type: none"> <li>Begriffsdefinition: Endlicher Automat (Akzeptor) als 5-Tupel</li> <li>Zustandsgraphen und Zustandstabellen</li> </ul>	<ul style="list-style-type: none"> <li>Ausgangspunkt: Wie kann analysiert werden, ob ein formalsprachlicher Ausdruck syntaktisch korrekt ist? → Gesucht sind effektive Algorithmen</li> <li>Für jede Tokenklasse kann ein Akzeptor angegeben werden</li> <li>Darstellung von EA mithilfe von Zustandsgraphen</li> </ul>	<ul style="list-style-type: none"> <li>EA können leicht als Computerprogramm realisiert werden.</li> <li>Arbeitsweise der Automaten sollte mithilfe einer Programmiersprache simuliert werden</li> <li>Nutzung von Modellierungswerkzeugen (z. B. AutoEdit)</li> </ul>
Kontextfreie Sprachen	<ul style="list-style-type: none"> <li>Aufbau der Regeln einer kontextfreien Grammatik</li> <li>Backtracking</li> </ul>	<ul style="list-style-type: none"> <li>Erkenntnis: Klammerstrukturen können nicht von EA erkannt werden. Diese lassen sich mit linkslinearen Sprachen nicht realisieren (anbn)</li> <li>Erzeugung von Klammerstrukturen durch kontextfreie Grammatiken</li> <li>Anwendung des Backtracking-Verfahrens bei der Syntaxanalyse (Parser)</li> </ul>	<ul style="list-style-type: none"> <li>Erläuterung der Arbeitsweise eines Parsers am Beispiel von „Mini-PASCAL“ oder ähnlichen einfachen Sprachkonstrukten</li> </ul>
Kellerautomat	<ul style="list-style-type: none"> <li>Definition des Kellerautomaten als 7-Tupel</li> <li>Kellerspeicher mit seinen begrenzten Zugriffsmöglichkeiten</li> </ul>	<ul style="list-style-type: none"> <li>Ausgehend von den „Schwächen“ des EA diskutieren die Schüler notwendige Veränderungen im Aufbau → Kellerspeicher</li> </ul>	<ul style="list-style-type: none"> <li>evtl. Verweis auf ADT</li> </ul>
Kontextsensitive Sprachen  TURING-Maschine	<ul style="list-style-type: none"> <li>Aufbau der Regeln einer kontextsensitiven Grammatik</li> <li>Definition der TM als 7-Tupel</li> </ul>	<ul style="list-style-type: none"> <li>Erkenntnis: Es gibt Elemente höherer Programmiersprachen, die nicht durch kontextfreie Grammatiken erzeugt werden können (anbncn).</li> <li>Schwächen des Kellerautomaten: Begrenzter Speicherzugriff → Abhilfe: wahlfreier Speicherzugriff → TM</li> <li>Anwendung der TM zum Erkennen einer Sprache der Form anbncn</li> </ul>	<ul style="list-style-type: none"> <li>Kurzvortrag zum Leben von und Wirken von Alan M. Turing</li> </ul>

<b>Schwerpunkte der UE (Zeit)</b>	<b>Inhalte/Begriffe</b>	<b>methodisch/didaktische Hinweise</b>	<b>Kontext (Erfahrungswelt der Schüler, weitere Themen, andere Fächer)</b>
Transduktoren	<ul style="list-style-type: none"> <li>• Modellierung von Sachverhalte aus dem täglichen Leben mithilfe eines Automaten und Abbildung der Überföhrungsfunktionen als gerichteter Graph</li> </ul>	<ul style="list-style-type: none"> <li>• Schüler verwenden selbstständig Modellierungswerkzeuge oder Programmierwerkzeugen</li> </ul>	<ul style="list-style-type: none"> <li>• Projekt in Gruppenarbeit möglich</li> </ul>
Präzisierung des Algorithmusbegriffs mithilfe der TM Grenzen der Berechenbarkeit	<ul style="list-style-type: none"> <li>• Beispiele für TM (Addier-Maschine, Kopier-Maschine)</li> <li>• Vergleich TM - Computer</li> <li>• Präzisierung des Algorithmusbegriffs</li> <li>• Grenzen der Berechenbarkeit</li> </ul>	<ul style="list-style-type: none"> <li>• Schüler nutzen Ablaufprotokolle und Zustandsgraphen zur Analyse der Arbeitsweise einer TM.</li> <li>• Konstruieren TM für konkrete Aufgabenstellungen</li> </ul>	<ul style="list-style-type: none"> <li>• Simulation von TM mithilfe einer höheren Programmiersprache</li> <li>• Ausblick und Vergleich zur Registermaschine</li> </ul>

## Testaufgabe

Gegeben ist der folgende endliche Automat (Akzeptor):

$A = (X, Z, \delta, z_o, Z_E)$  mit

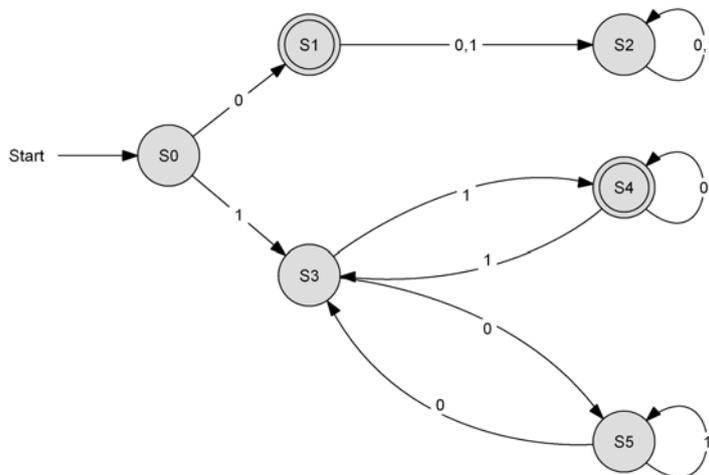
$X = \{ 0, 1 \}$ ,

$Z = \{ s_0, s_1, s_2, s_3, s_4, s_5 \}$ ,

$z_o = s_0$ ,

$Z_E = \{ s_1, s_4 \}$ ,

$\delta$  ist durch den folgenden Zustandsgraphen gegeben:



Aufgabe	Abschlussorientierter Standard
1. Stellen Sie natürliche und formale Sprachen gegenüber. Gehen Sie dabei auf deren Bedeutung, Gemeinsamkeiten und Unterschiede ein.	vergleichen formale und natürliche Sprachen unterscheiden zwischen Syntax und Semantik
2. Geben Sie jeweils zwei Beispiele für Worte an, die von A akzeptiert bzw. nicht akzeptiert werden.	analysieren formale Sprachen
3. Beschreiben Sie formal die Sprache $L(A)$ , die A akzeptiert. <b>Hinweis:</b> Fassen Sie die Worte als Dualzahlen auf und wandeln Sie diese in Dezimalzahlen um.	analysieren formale Sprachen
4. Geben Sie eine Grammatik $G = (V_N, V_T, P, S)$ an, die die gleiche Sprache $L_A$ erzeugt.	Beschreiben beispielhaft den Zusammenhang zwischen Automaten und Grammatiken
5. Erläutern Sie Aufbau und Arbeitsweise einer TURING-Maschine. Definieren Sie eine TURING-Maschine T, die die gleiche Aufgabe realisiert wie dieser Akzeptor. Die Überföhrungsfunktion von T ist als Zustandsgraph anzugeben.	Analysieren den Aufbau und die Arbeitsweise eines allgemeinen Maschinenmodells Entwickeln einfache Modelle

## 4 Sprachen und Automaten (GK)

	<b>Eingangsvoraussetzungen Die Schülerinnen und Schüler</b>	<b>Abschlussorientierte Standards Die Schülerinnen und Schüler</b>
<b>Informatisches Modellieren</b>  <i>Modelle erstellen und bewerten</i>	<ul style="list-style-type: none"> <li>• beschreiben Modelle als vereinfachtes Abbild der realen Welt,               <ul style="list-style-type: none"> <li>○ kennen das Konzept der OOM bei Standardanwendungen</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• analysieren Realitätsausschnitte und wählen ein geeignetes Modellierungsverfahren aus,               <ul style="list-style-type: none"> <li>○ modellieren mithilfe des Modells Endlicher Automat,</li> </ul> </li> <li>• variieren und erweitern vorgegebene Modelle,               <ul style="list-style-type: none"> <li>○ entwickeln, implementieren und testen mit einer höheren Programmiersprache oder mit geeigneten Tools (z.B. Automaten-Kara)</li> </ul> </li> <li>• reflektieren und beurteilen die eigene Modellierung,               <ul style="list-style-type: none"> <li>○ überprüfen an Hand von Ablaufprotokollen bzw. Simulationsprogrammen die korrekte Arbeitsweise der Automatenmodelle</li> </ul> </li> <li>• erläutern Basiskonzepte der zustandsorientierten Modellierung,               <ul style="list-style-type: none"> <li>○ stellen die Überföhrungsfunktion als Zustandsgraph dar</li> </ul> </li> <li>• modellieren automatisierte Abläufe mit Hilfe endlicher Automaten               <ul style="list-style-type: none"> <li>○ zur Analyse formaler Sprachen. (z. B. Akzeptoren )</li> <li>○ zur Abbildung Endlicher Automaten mit Ausgabe (z .B. Transduktoren)</li> </ul> </li> </ul>
<b>Mit Information umgehen</b>  <i>Information in Form von Daten darstellen und verarbeiten</i>		<ul style="list-style-type: none"> <li>• unterscheiden zwischen Syntax und Semantik und erläutern dies in natürlichen, halbformalen und formalen Sprachen.               <ul style="list-style-type: none"> <li>○ erläutern die Begriffe Syntax und Semantik und arbeiten diesbezüglich Gemeinsamkeiten und Unterschiede von natürlichen, halbformalen und formalen Sprachen heraus</li> </ul> </li> </ul>
<b>Informatiksysteme verstehen</b>  <i>Wirkprinzipien kennen und anwenden</i>	<ul style="list-style-type: none"> <li>• erläutern Eigenschaften von Algorithmen an einfachen Beispielen,               <ul style="list-style-type: none"> <li>○ kennen Eigenschaften von Algorithmen aus der Arbeit mit Standardanwendungen und verschiedenen Programmierwerkzeugen und aus dem täglichen Leben</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• vergleichen formale und natürliche Sprachen,               <ul style="list-style-type: none"> <li>○ kennen Gemeinsamkeiten (Alphabet, Syntax, Semantik) und Unterschiede (Sprachgefüge, zeitliche Veränderlichkeit von Vokabular und Grammatik) der einzelnen Sprachen</li> </ul> </li> </ul>

<p><b>Problemlösen</b> <i>Probleme erfassen und mit Informatiksystemen lösen</i></p>	<ul style="list-style-type: none"> <li>• beschreiben algorithmische Abläufe umgangssprachlich und grafisch, <ul style="list-style-type: none"> <li>○ formulieren Algorithmen verbal und arbeiten mit Struktogrammen und/oder PAP</li> </ul> </li> <li>• modellieren einfache Abläufe mit Algorithmen (Sequenz, Auswahl, Wiederholung), <ul style="list-style-type: none"> <li>○ sind in der Lage Vorgänge aus dem täglichen Leben bzw. mathematische Berechnungen algorithmisch zu modellieren</li> </ul> </li> <li>• setzen Algorithmen in Programme um. <ul style="list-style-type: none"> <li>○ arbeiten selbstständig mit ausgewählten Werkzeugen</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• setzen informatische Methoden(z. B. Bottom-Up,Top-Down) bei der Syntaxanalyse ein <ul style="list-style-type: none"> <li>○ an kleinen Ausschnitten von natürlichen Sprachen oder Programmiersprachen erfolgt die syntaktische Analyse mithilfe der Bottom-Up- bzw.Top-Down-Methode</li> </ul> </li> <li>• beachten Grenzen des Problemlösens mit Informatiksystemen <ul style="list-style-type: none"> <li>○ kennen die Grenzen des Problemlösens am Beispiel des Halteproblems</li> </ul> </li> </ul>
<p><b>Kommunizieren und kooperieren</b> <i>Teamarbeit organisieren und koordinieren</i></p>	<ul style="list-style-type: none"> <li>• verwenden im angemessenen Rahmen die Fachsprache,</li> <li>• dokumentieren und präsentieren ihre Arbeitsergebnisse.</li> </ul>	<ul style="list-style-type: none"> <li>• verfügen über eine angemessene Fachsprache und verwenden sie sachgerecht, <ul style="list-style-type: none"> <li>○ verwenden und kennen die Begriffe Alphabet, Wort, Wortmenge, Sprache, Syntax, Semantik, Endlicher Automat</li> </ul> </li> </ul>
<p><b>Wechselwirkung zwischen Informatiksystemen, Mensch und Gesellschaft beurteilen</b> <i>Anwendungen erfassen und Auswirkungen abschätzen</i></p>		<ul style="list-style-type: none"> <li>• bewerten Risiken und Chancen von Informatiksystemen <ul style="list-style-type: none"> <li>○ Auswirkungen des Einsatzes von Automaten auf die Arbeitswelt</li> </ul> </li> </ul>

## Zeitlicher Verlauf und methodische Hinweise

Schwerpunkte der UE (Zeit)	Inhalte/Begriffe	methodisch/didaktische Hinweise	Kontext (Erfahrungswelt der Schüler, weitere Themen, andere Fächer)
Klassifikation von Sprachen	<ul style="list-style-type: none"> <li>• Unterschiede und Gemeinsamkeiten natürlicher und künstlicher Sprachen Grammatik, Syntax, Semantik, Pragmatik</li> </ul>	<ul style="list-style-type: none"> <li>• Schüler erarbeiten die Definitionen der Begriffe unter Einbeziehung ihres Wissens aus dem Deutsch- und Fremdsprachenunterricht und erläutern diese an Beispielen</li> </ul>	<ul style="list-style-type: none"> <li>• Schüler kennen natürliche Sprachen (Deutsch, Englisch, Französisch...) Programmiersprachen und die mathematische Formelsprache</li> </ul>
Beschreibungen formaler Sprachen	<ul style="list-style-type: none"> <li>• Sprachen für die Mensch-Maschine-Kommunikation müssen für die automatische Verarbeitung geeignet sein</li> <li>• Syntaxdiagramme,</li> <li>• Bottom up- und Topdown-Ableitung</li> <li>• Ableitungsbaum</li> </ul>	<ul style="list-style-type: none"> <li>• Ausgangspunkt: Wie muss eine Sprache beschaffen sein, damit sie für die Mensch-Maschine-Kommunikation geeignet ist?</li> <li>• Übertragung dieser Erkenntnisse auf Ausschnitte von natürlichen Sprachen und Überleitung zu einer bekannten höheren Programmiersprache</li> <li>• Darstellung der Sprachregel mithilfe von Syntaxdiagrammen</li> <li>• Schüler weisen durch Ableitungen nach, ob ein zur Sprache gehörendes Wort (Satz) vorliegt.</li> </ul>	<ul style="list-style-type: none"> <li>• Schüler stellen immer einen Bezug zur deutschen Sprache (bzw. Fremdsprache) her</li> </ul>
Endliche Automaten (Akzeptor)	<ul style="list-style-type: none"> <li>• Begriffsdefinition: Endlicher Automat (Akzeptor)</li> <li>• Aufbau und Arbeitsweise EA</li> <li>• Zustandsgraphen</li> </ul>	<ul style="list-style-type: none"> <li>• Ausgangspunkt: Wie kann analysiert werden, ob ein formalsprachlicher Ausdruck syntaktisch korrekt ist? → Gesucht sind effektive Algorithmen</li> <li>• Darstellung von EA mithilfe von Zustandsgraphen</li> </ul>	<ul style="list-style-type: none"> <li>• Aufgaben eines Compilers: Erkennung der einzelnen Anweisungen und Übertragung in Maschinensprache</li> <li>• EA können leicht als Computerprogramm realisiert werden.</li> <li>• Arbeitsweise der Automaten sollte mithilfe einer Programmiersprache simuliert werden</li> <li>• Nutzung von Modellierungswerkzeugen (z. B. AutoEdit)</li> </ul>
Transduktoren	<ul style="list-style-type: none"> <li>• Modellierung von Sachverhalte aus dem täglichen Leben mithilfe eines Automaten und Abbildung der Überföhrungsfunktion als gerichteter Graph</li> </ul>	<ul style="list-style-type: none"> <li>• Schüler verwenden selbstständig Modellierungswerkzeuge oder Programmierwerkzeugen</li> </ul>	<ul style="list-style-type: none"> <li>• Projekt in Gruppenarbeit möglich</li> </ul>



