



# Hans-Brüggemann-Schule

Gemeinschaftsschule mit Oberstufe  
des Schulverbandes Bordsesholm in Bordsesholm

## Fachschaft Informatik

### Schulinternes Fachcurriculum Informatik – Sekundarstufe II

Stand: 01.05.2025

beschlossen von der Fachkonferenz Informatik am xx.xx.2025

#### Präambel

Das vorliegende schulinterne Fachcurriculum für das Fach Informatik in der gymnasialen Oberstufe basiert auf den aktuellen Fachanforderungen Informatik Sekundarstufe II des Landes Schleswig-Holstein (veröffentlicht 2021) sowie dem zugehörigen Leitfaden zur Umsetzung der Fachanforderungen. Es überträgt die verbindlichen landesweiten Kompetenzziele, Inhalte und didaktischen Hinweise in eine schulbezogene Planung und bietet damit eine strukturierte Grundlage für den Informatikunterricht in der Einführungs- und perspektivisch auch in der Qualifikationsphase.

Aktuell bezieht sich dieses Curriculum ausschließlich auf die Einführungsphase (E-Phase). Die Konkretisierung der Inhalte und Kompetenzen für die Qualifikationsphase (Q1 und Q2) erfolgt aufwachsend in den kommenden Schuljahren, in Anlehnung an die Fachanforderungen und unter Einbeziehung der Erfahrungen aus der E-Phase.

Das Curriculum dient der Transparenz und Orientierung: für Lehrkräfte, Schülerinnen und Schüler sowie die Schulöffentlichkeit. Es beschreibt einen Ist-Zustand der curricularen Umsetzung an unserer Schule zum Zeitpunkt der letzten Überarbeitung. Da sich sowohl gesellschaftliche und technologische Entwicklungen als auch die Zusammensetzung und Bedarfe von Lerngruppen fortlaufend verändern, wird dieses Curriculum als dynamisches Dokument verstanden. Es wird regelmäßig im Rahmen von Fachkonferenzen überprüft und bei Bedarf weiterentwickelt.

Die im Curriculum dargestellten Themen, Inhalte und methodischen Vorschläge stellen daher keine starren Vorgaben, sondern Orientierungshilfen dar. Die Verantwortung für die konkrete unterrichtliche Umsetzung liegt bei den jeweiligen Lehrkräften. Sie entscheiden auf Grundlage des vorliegenden Curriculums, der gültigen Fachanforderungen sowie der individuellen Voraussetzungen der Lernenden über Schwerpunktsetzungen, methodisch-didaktische Zugänge und Differenzierungsmaßnahmen.

Ziel bleibt es, den Schülerinnen und Schülern einen verständnisorientierten, lebensweltlich relevanten und zukunftsgerichteten Informatikunterricht zu ermöglichen, der sowohl grundlegende Kompetenzen im Umgang mit digitalen Technologien vermittelt als auch zur

aktiven, kritischen und kreativen Auseinandersetzung mit informatischen Systemen und ihren gesellschaftlichen Auswirkungen befähigt.

## **Grundsätzliche Absprachen zur Unterrichtsgestaltung**

### ***Fachsprache und Darstellungsformen***

Fachbegriffe werden grundsätzlich in deutscher und englischer Sprache angegeben, beispielsweise „Zeichenkette (String)“ oder „Schleife/Wiederholung (loop)“. Ist im fachlichen Diskurs ausschließlich die englischsprachige Bezeichnung etabliert, so wird diese verwendet. Für die Darstellung von Programmabläufen werden Flowcharts (Flussdiagramme) und Blockdiagramme genutzt. Fachtermini, die bereits in der Sekundarstufe I eingeführt wurden, werden in der Sekundarstufe II konsequent weiterverwendet (z. B. der Begriff „Wiederholung“ aus Scratch findet seine Fortsetzung in der Arbeit mit Python).

### ***Binnendifferenzierung: Fordern und Fördern***

Projektorientierte Unterrichtssequenzen bieten ein hohes Maß an innerer Differenzierung. Aufgabenstellungen werden differenziert für Lernende mit unterschiedlichen Vorkenntnissen (Anfänger/Fortgeschrittene) gestaltet. Zusätzlich werden leistungsstarke Schülerinnen und Schüler gezielt gefördert, etwa durch die Teilnahme an Wettbewerben wie „Project Euler“, dem Bundeswettbewerb Informatik oder dem Informatik-Biber. Auch die Teilnahme an der Software Challenge in Kleingruppen (3–5 Lernende) wird empfohlen. Interessierte Lernende können motiviert werden, zusätzlich eine weitere Programmiersprache zu erlernen oder zu erproben. Für bestimmte Wettbewerbe können ggf. auch kursübergreifende Gruppen gebildet werden. In spielerisch-kreativen Formaten wie Programmierturnieren („Core Wars“, „Tit for Tat“) kann informatisches Denken vertieft werden. Darüber hinaus kann z. B. die Teilnahme an einem Schnupperstudium empfohlen.

Auch weniger geübte Schülerinnen und Schüler werden durch binnendifferenzierte Aufgabenformate gezielt unterstützt. Es wird vor allem berücksichtigt, dass aktuell noch nicht alle Lernenden der Sekundarstufe II auch in der Sekundarstufe I schon Informatikunterricht hatten und es erfahrungsgemäß viele neue Schülerinnen und Schüler von extern in der E-Phase gibt. Die Fachschaft entwickelt daher bedarfsgerechte Fördermaterialien für die jeweiligen Jahrgangsstufen.

Außerunterrichtliche Unterstützungsangebote werden aktiv empfohlen: etwa Erklärvideos oder interaktive Programmierübungen.

Wettbewerbe, die Initiative „Rent a Scientist“ (Wissenschaftlerinnen und Wissenschaftlicher als Role Models), Girls’ und Boys’ Days, die gezielte Kooperation mit der Wirtschaft und das Knüpfen von Kontakten bei Berufsorientierungsmessen (z. B. BOBB) werden können Eingang in den Informatikunterricht finden.

### ***Digitale Medien und Medienkompetenz***

Im Rahmen des Informatikunterrichts wird ein Beitrag zur Medienkompetenzbildung geleistet. Schülerinnen und Schüler nutzen unter anderem GitHub als Werkzeug zur Versionskontrolle und Kollaboration. Sie erstellen regelmäßig digitale Produkte (Texte, Medieninhalte) unter Einhaltung vorgegebener Formatkonventionen. Der kompetente Umgang mit gleichartigen Anwendungen aus dem Bereich Medienproduktion (z. B. Video- und Audioschnitt) wird ebenso gefördert wie die Arbeit mit Unix- und Linux-Systemen, etwa in Form virtualisierter Umgebungen.

### **Hilfsmittel**

Der Einsatz von digitalen Werkzeugen ist fester Bestandteil des Unterrichts. Zum Einsatz kommen Computer und Laptops zur Programmierung, zur Arbeit an strukturierten Textdokumenten sowie zur digitalen Mediengestaltung. Als Mikrocontrollersysteme dienen u. a. Calliopes und Arduinos (ggf. von den Lernenden selbst angeschafft), ergänzt durch schulische Materialien wie Kabel, LEDs, Sensoren etc. Auch Lego-Roboter oder 3-D-Drucker können genutzt werden. Weitere Hilfsmittel sind klassische Hefter oder digitale Notizen auf Tablets/Laptops sowie gängige Textverarbeitungsprogramme (z. B. LibreOffice Writer).

### **Leistungsbewertung**

Die Leistungsbewertung erfolgt auf Grundlage der aktuellen Erlasslage. Pro Halbjahr wird eine Klausur geschrieben. Diese darf kurze Programmieraufgaben beinhalten und bei Bedarf auch in digitaler Form durchgeführt werden. Alternativ kann eine der beiden Klausuren durch eine gleichwertige Leistung ersetzt werden (z. B. ein umfangreicher Projektbericht mit Dokumentation), sofern sie hinsichtlich des Umfangs, der Vorbereitung und der Reflexion gleichwertig ist. Alle Leistungsnachweise müssen Aufgaben aus den drei Anforderungsbereichen enthalten. Die schriftlichen oder gleichwertigen Leistungsnachweise tragen 30%, sämtliche anderen erbrachten (Unterrichts-)Beiträge 70% zur Gesamtnote bei.

### **Überprüfung und Weiterentwicklung**

Das schulinterne Fachcurriculum wird regelmäßig im Rahmen von Fachkonferenzen evaluiert. Dabei werden sowohl die Angemessenheit der Inhalte als auch aktuelle Herausforderungen und Bedarfe reflektiert. Daraus ergeben sich ggf. Anpassungen und Weiterentwicklungen des Curriculums. Gegenseitige Hospitationen mit anschließender kollegialer Rückmeldung werden als zusätzliches Instrument der Qualitätssicherung und Curriculumentwicklung empfohlen.

### Thematische Struktur Einführungsphase – Kontexte, Kompetenzen und Umsetzungshinweise

Die nachfolgende Tabelle bietet einen strukturierten Überblick über zentrale Themenbereiche des Informatikunterrichts in der Einführungsphase. Sie verknüpft bedeutsame Leitfragen mit fachlichen Inhalten und Kompetenzen sowie praxisorientierte Hinweise zur Umsetzung im Unterricht.

Leitfrage	Mögliche Unterrichtsinhalte	Inhaltsbezogene Kompetenzen <i>Die Schülerinnen und Schüler...</i>	Hinweise zur Umsetzung/Tipps/Material
<b>Wie denkt ein Computer?</b>	<ul style="list-style-type: none"> <li>- Begriff Algorithmus (als endliche Beschreibung von effektiv ausführbaren Arbeitsschritten)</li> <li>- einfacher Pseudocode</li> </ul>	<p>... beschreiben sowohl für einzelne Anweisungen als auch für Algorithmen im Ganzen das Ergebnis der Ausführung.</p> <p>... formulieren Handlungsvorschriften unter Nutzung algorithmischer Grundbausteine</p>	<p>Flowcharts, Pseudocode</p> <p>Projektaufgabe: „Alltag als Algorithmus“</p> <p>Einstieg Calliope (blockbasiert)</p> <p><a href="https://inf-schule.de/algorithmen">https://inf-schule.de/algorithmen</a></p>
<b>Wie speichert ein Computer Informationen?</b>	<ul style="list-style-type: none"> <li>- Bits und Bytes</li> <li>- Binär- und Hexadezimalsystem</li> <li>- Codierung von Zahlen und Zeichen; Wahrheitswerte</li> <li>- Codierung: Binär, ASCII, Unicode</li> <li>- Darstellungen: ganze Zahlen, Fließkommazahlen</li> </ul>	<p>... beschreiben die binäre Repräsentation von Daten.</p> <p>... überführen Informationen in eine Datenrepräsentation im Speicher.</p>	<p>Rechenübungen, QR-Codes selbst erstellen</p> <p>ASCII-Kunst als kreativer Zugang</p> <p>Codierungsspiele z. B. mit Emojis oder Geheimschrift</p> <p><a href="https://inf-schule.de/information">https://inf-schule.de/information</a></p>
<b>Wie spricht man mit Maschinen?</b>	<ul style="list-style-type: none"> <li>- elementare Anweisung</li> <li>- Sequenz</li> <li>- Kontrollstrukturen und ihre Verschachtelung</li> <li>- Variablen, Bezeichner, Datentyp, Definition, Zuweisung</li> <li>- algorithmische Konzepte (einfache Iteration, warten auf ein Ergebnis, verschachtelte Wiederholungen, erschöpfende Suche, Aufzählen und Testen)</li> <li>- Rollen von Variablen in Programmtexten (Zählvariable, Wertspeicher, Akkumulator, Indikator, Index)</li> <li>- Modularisierung</li> <li>- Parameter und Argument</li> <li>- Rückgabewert</li> <li>- Suchalgorithmen, Sortieralgorithmen</li> </ul>	<p>... interpretieren und kommentieren einfache Algorithmen (in einer geeigneten Programmierumgebung).</p> <p>... entwerfen und implementieren Algorithmen zur Lösung einer gegebenen Problemstellung.</p> <p>... erläutern Standardalgorithmen.</p>	<p>Code.org</p> <p>Digitale Fehlersuche-Trainings</p> <p>Fehlersammelheft („Bugbook“)</p> <p>Reflexion über Fehlertypen (Syntax, Logik, Laufzeit)</p> <p><a href="https://inf-schule.de/imperative-programmierung">https://inf-schule.de/imperative-programmierung</a></p>

<p><b>Wie arbeiten viele Menschen gemeinsam an Software?*</b></p>	<ul style="list-style-type: none"> <li>- Versionskontrolle mit Git, Einführung in GitHub</li> <li>- Dokumentation mit Markdown</li> <li>- Versionsverwaltung</li> <li>- Kooperative Programmierung</li> <li>- Einfache Dokumentationsstandards</li> </ul>	<p>... gestalten reflektiert die Zusammenarbeit bei der Softwareentwicklung mithilfe digitaler Werkzeuge.</p>	<p>Projektarbeit in Gruppen Gemeinsames GitHub-Repo mit Change Logs Erstellen von Readme-Dateien</p>
<p><b>Was steckt im Inneren eines Computers?</b></p>	<ul style="list-style-type: none"> <li>- Sensoren und Eingabegeräte, Prozessoren, Speicher, Netzwerkkomponenten, Aktoren und Ausgabegeräte</li> <li>- Von-Neumann-Maschinenmodell: Steuerwerk, Rechenwerk, Speicherwerk, Fetch-Zyklus</li> <li>- Ressourcenverwaltung (Datei-, Speicher-, Prozess-, Benutzer-, Rechteverwaltung)</li> </ul>	<p>... erklären die grundlegende Funktionsweise von Hardwarekomponenten und deren Zusammenwirken. ... beschreiben die Funktion und das Zusammenwirken von Rechen-, Steuer- und Speicherwerk in einem Von-Neumann-Rechner. ... beschreiben die Funktionen eines Betriebssystems.</p>	<p>Animationen, Simulationen (z. B. <a href="https://www.inf-schule.de">https://www.inf-schule.de</a>) Hardwaremodelle, z. B. PC-Demogeräte oder Online-Demos <a href="https://inf-schule.de/rechner">https://inf-schule.de/rechner</a></p>
<p><b>Kann ein Computer denken?</b></p>	<ul style="list-style-type: none"> <li>- Grundlagen KI, Entscheidungsbäume, neuronale Netze (vereinfacht)</li> <li>- Forward Propagation</li> <li>- Multi-Layer-Perzeption</li> <li>- Trainings- und Validierungsdaten</li> <li>- Chancen &amp; Risiken</li> <li>- überwachtes Lernen, unüberwachtes Lernen, Verstärkungslernen</li> <li>- Anwendungsszenarien (z. B. Chatbots, Bilderkennung)</li> <li>- Ethische Fragen und Datenschutzaspekte</li> </ul>	<p>... analysieren einfache künstliche neuronale Netze. ... erläutern typische Strategien maschinellen Lernens.</p>	<p>Projekte zu „Fairer KI“ KI unplugged (z. B. decision tree activities) Diskussion: „Kann KI kreativ sein?“ <a href="https://inf-schule.de/ki">https://inf-schule.de/ki</a></p>
<p><b>Wie schützt man sich vor Gefahren im Netz?</b></p>	<ul style="list-style-type: none"> <li>- Sicherheitsziele: Geheimhaltung, Nachrichten- und Teilnehmerauthentizität, Integrität</li> <li>- Risiken: Man-in-the-middle, Phishing, Viren und Trojaner</li> <li>- Abwehrmaßnahmen: Firewall, sicherer Kanal, Public-Key-Infrastruktur, Virens Scanner</li> <li>- Benutzername und Passwort</li> <li>- Zertifikate</li> </ul>	<p>... diskutieren Sicherheitsrisiken und Abwehrmaßnahmen in Netzwerken. ... beschreiben und beurteilen Authentifizierungsverfahren. ... diskutieren die Folgen netzbasierter Kommunikation für Entwicklungen in Politik und Gesellschaft.</p>	<p>Sicherheitschecks am eigenen Gerät Erstellen sicherer Passwörter Planspiel: „IT-Notfall an der Schule“ <a href="https://inf-schule.de/kryptologie">https://inf-schule.de/kryptologie</a> <a href="https://inf-schule.de/gesellschaft">https://inf-schule.de/gesellschaft</a></p>

	<ul style="list-style-type: none"><li>- biometrische Authentifizierung</li><li>- Zwei-Faktor-Authentifizierung</li><li>- soziale Netzwerke</li><li>- sichere Kommunikation</li><li>- Manipulation und Meinungsbildung</li><li>- Anonymität im Internet</li></ul>		
--	--	--	--

*\*optional, für das grundlegende Anforderungsniveau nicht in den Fachanforderungen enthalten*

Zu den in der Tabelle genannten Leitfragen lassen sich beliebig viele alltagsnahe Fragestellungen nennen, die sich eignen, einen Zugang zu dem entsprechenden Inhaltsbereich zu finden. Nachfolgend sind einige dieser Fragen genannt, letztlich liegt es aber immer in der Hand der Lehrkraft, abhängig von den Vorkenntnissen und Interessen der Lerngruppe eine angemessene Kontextualisierung vorzunehmen.

- **1. Wie denkt ein Computer?**

- Wie kann man einem Staubsaugerroboter beibringen, eine Wohnung effizient zu reinigen?
- Warum funktioniert ein Fahrstuhl immer in einer bestimmten Reihenfolge?
- Wie erkläre ich einer Maschine, wie man einen Kaffee kocht?
- Was passiert, wenn ich bei Google etwas falsch schreibe – und es trotzdem erkannt wird?
- Wie unterscheiden sich menschliches und maschinelles Denken?
- Warum scheitert ein Automat manchmal an ungewöhnlichen Situationen?
- Wie programmiere ich eine Spielesequenz mit Start, Regeln und Ende?
- Warum wiederholen sich bestimmte Prozesse in Apps immer gleich?

- **2. Wie spricht man mit Maschinen?**

- Was bedeutet eine Fehlermeldung beim Programmieren?
- Warum verstehen Maschinen keine Umgangssprache?
- Wie kann man Code für andere Menschen lesbar schreiben?
- Wie unterscheidet ein Computer Eingabe- von Ausgabebefehlen?
- Wie schreibe ich so, dass mein Code keine unvorhergesehenen Fehler produziert?
- Wie kann man erkennen, wo ein Fehler im Code steckt?
- Warum stürzt ein Programm bei einem kleinen Eingabefehler ab?
- Wie hilft Kommentieren im Code dem Verständnis?
- Wie kann ich einem Roboter erklären, was er tun soll?

- **3. Wie speichert ein Computer Informationen?**

- Wie wird ein Foto digital gespeichert?
- Was passiert, wenn ich ein Bild komprimiere?
- Wie speichert mein Handy Emojis oder Buchstaben?
- Warum ist mein Speicher plötzlich voll?
- Wie funktionieren QR-Codes?
- Wie speichert ein USB-Stick Daten?

- Was passiert beim Speichern eines Word-Dokuments?
- Warum ist eine Datei beschädigt und nicht mehr lesbar?
- **4. Was steckt im Inneren eines Computers?**
- Was passiert, wenn ich den Ein-Knopf am PC drücke?
- Wie gelangt ein Tastendruck zur Anzeige auf dem Bildschirm?
- Was macht ein Prozessor den ganzen Tag?
- Was ist RAM, und warum ist er wichtig?
- Warum macht mein Laptop Geräusche, wenn er arbeitet?
- Wie unterscheidet sich der Arbeitsspeicher vom Langzeitspeicher?
- Was passiert bei einem Stromausfall mitten in der Arbeit?
- Wie arbeitet ein Scanner oder Drucker mit dem Computer zusammen?
- Wie wird ein PC eigentlich langsamer mit der Zeit?
- **5. Was passiert, bevor ein Programm startet?**
- Wie funktioniert die Elektronik in einem Lichtschalter?
- Warum kann man mit 0 und 1 rechnen?
- Wie funktionieren Schaltkreise in einem Taschenrechner?
- Wie addiert ein Computer zwei Zahlen?
- Was machen Logikgatter?
- Wie kann eine Maschine zählen?
- Wie kann man mit Strom logisch denken?
- Was ist der Unterschied zwischen analog und digital?
- Wie merkt sich ein Gerät den aktuellen Zustand?
- **6. Wie arbeiten viele Menschen gemeinsam an Software?**
- Wie kann ein Team gleichzeitig an einer App arbeiten?
- Was ist der Unterschied zwischen lokalem Speichern und Cloud-Arbeit?
- Wie erkennt man, wer was am Code verändert hat?
- Wie vermeidet man, dass sich Codeversionen überschneiden?
- Was ist ein Commit und warum ist er wichtig?
- Wie arbeitet ein Entwicklerteam in einem großen Unternehmen zusammen?
- Wie dokumentiert man Veränderungen im Code?
- Wie organisiert man Arbeit an einem Schulprojekt digital?

- **7. Kann ein Computer denken?**

- Wie erkennt Instagram mein Gesicht auf Fotos?
- Kann ein Computer Gefühle erkennen?
- Warum weiß Spotify, welche Musik mir gefällt?
  
- Wie unterscheiden sich Mensch und KI im Denken?
- Was macht eine KI „intelligent“?
- Kann eine KI kreativ sein?
- Warum ist ChatGPT manchmal falsch, obwohl es „schlau“ wirkt?
- Wie lernen Maschinen, ohne dass man sie programmiert?
- Was sind ethische Fragen beim Einsatz von KI?
- Wie beeinflusst KI mein Leben heute schon?

- **8. Wie schützt man sich vor Gefahren im Netz?**

- Woran erkenne ich Phishing-Mails?
- Wie sicher ist mein WLAN-Passwort wirklich?
- Warum sollte ich für jede Website ein anderes Passwort haben?
- Wie schützt mich eine Firewall?
- Wie kann ich meine Daten beim Surfen anonym halten?
- Was bedeutet Ende-zu-Ende-Verschlüsselung?
- Wie erkenne ich Fake-Apps im App-Store?
- Was tun, wenn mein Social-Media-Konto gehackt wurde?
- Warum will jede App auf mein Mikrofon zugreifen?