Vielfältig. Wie Du.

Schulinterner Lehrplan – Sek II

Informatik

Städtische
Gesamtschule Kaarst-Büttgen
Hubertusstraße 22 - 24
41564 Kaarst
Tel.: 02131 – 20 27 512

Inhaltsverzeichnis

1 Rahmenbedingungen der fachlichen Arbeit 3
2 Entscheidungen zum Unterricht 4
2.1 Unterrichtsvorhaben 4
2.1.1 Übersichtsraster Unterrichtsvorhaben 5
Einführungsphase 5
Qualifikationsphase 1 7
Qualifikationsphase 2 9

2.1.2 Konkretisierte Unterrichtsvorhaben 10
Einführungsphase 10
Unterrichtsvorhaben EF – I 10
Unterrichtsvorhaben EF - II 11
Unterrichtsvorhaben EF - III 13
Unterrichtsvorhaben EF - IV 14

Qualifikationsphase 15
Unterrichtsvorhaben Q 1 – I 15
Unterrichtsvorhaben Q 1 - II 16
Unterrichtsvorhaben Q 1 - III 16
Unterrichtsvorhaben Q 1 - IV 17

Unterrichtsvorhaben Q 2 – I 18
Unterrichtsvorhaben Q 2 - II 20
Unterrichtsvorhaben Q 2 – III 21

2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit 22
2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung 23
2.3.1 Beurteilungsbereich Klausuren 23
2.3.2 Beurteilungsbereich Sonstige Mitarbeit 24
2.3.3 Grundsätze der Leistungsrückmeldung und Beratung 27
2.4 Lehr- und Lernmittel 27
3. Entscheidungen zu fachdidaktischen- und unterrichtsübergreifenden Fragen 28
4 Qualitätssicherung und Evaluation 29

1. Rahmenbedingungen der fachlichen Arbeit

Die Städtische Gesamtschule Kaarst - Büttgen wurde 2013 gegründet. Sie liegt im Grüngürtel des Stadtteils Büttgen und ist mit öffentlichen Verkehrsmitteln wie Bus und S-Bahn sowie dem Fahrrad zu erreichen.

Als einzige Gesamtschule in der Region Kaarst und Korschenbroich wollen wir eine Schule für alle Kinder sein, die gerne lernen, die sich Ziele setzen und sich zutrauen, dies im Laufe der kommenden Jahre motiviert umzusetzen.

Bei uns können alle Schulabschlüsse erreicht werden: vom Hauptschulabschluss nach der Klasse 9 bis zur Allgemeinen Hochschulreife - dem Abitur - nach der Klasse 13.

Mit zeitgemäßen pädagogischen Ansätzen werden die Schüler*innen auf einen geeigneten Weg ins Berufsleben oder in ein Studium vorbereitet. Ganz besonders wichtig ist uns die Förderung und Forderung unserer Schüler*innen. Dazu stehen uns als Instrumente nicht nur die Leistungsdifferenzierung in den Fächern Deutsch, Mathematik, Englisch und Chemie, zusätzliche Lernzeiten zum eigenverantwortlichen und selbstbestimmten Lernen, sondern auch eine breite Vielfalt an Fächern und außerunterrichtlichen Angeboten zur Verfügung.

Das Fach Informatik wird ab der Jahrgangsstufe 7 im Wahlpflichtbereich I (WP I) dreistündig unterrichtet. Ab Jahrgangsstufe 8 werden Informatik – Themen wie Programmieren mit Lego Mindstorms, mit dem Raspberry Pi, Calliope Mini und ein Kurs mit dem Schwerpunkt Smart Home im Wahlpflichtbereich II (WP II) angeboten. Dort wird auch auf die technische Informatik am Beispiel der Robotik eingegangen und mit außerschulischen Partner kooperiert.

Seit dem Schuljahr 2019/2020 führt die Gesamtschule Kaarst – Büttgen im ersten Jahrgang der Sekundarstufe II (S II) das Fach Informatik in einem Grundkurs durch.

Um insbesondere den Schüler*innen gerecht zu werden, die in der Sekundarstufe I nicht am Informatikunterricht teilgenommen haben, wird im Kurs der Einführungsphase kein Wert daraufgelegt, dass keine Vorkenntnisse für den erfolgreichen Durchlauf des Kurses notwendig sind.

Der Unterricht der Sekundarstufen wird mit der Programmiersprache Java durchgeführt. In der Einführungsphase kommt dabei das Programm Greenfoot mit Java - Standard zum Einsatz, die „Programmieranfängern ab 15 Jahren“ das Erstellen von grafischen Programmen erleichtert.

Zurzeit besteht die Fachschaft der Gesamtschule Kaarst aus 2 Kollegen, denen 2 Computerräume mit 17 bzw. 15 Rechnern zur Verfügung stehen. Alle Arbeitsplätze sind an das schulinterne Rechnernetz angeschlossen, so dass Schüler*innen über einen individuell

gestaltbaren Zugang zum zentralen Server der Schule alle Arbeitsplätze der zwei Räume zum Zugriff auf ihre eigenen Dateien, zur Recherche im Internet oder zur Bearbeitung schulischer Aufgaben verwenden können.
Der Unterricht erfolgt im 45-Minuten-Takt. Die Kursblockung sieht grundsätzlich für Grundkurse eine Doppelstunde und eine Einzelstunde vor.

2. Entscheidungen zum Unterricht

Implementation der 17 Ziele in den Lehrplan

Die im Jahre 2015 verabschiedeten Ziele der Weltgemeinschaft, die sogenannten ***SGD´***s (Sustain Development Goals) umfassen sowohl ökologische, ökonomische als auch soziale Aspekte bzw. Dimensionen im Hinblick auf die weitere Entwicklung der Welt (Agenda 2030[1]). Die Ziele richten sich nicht nur an Staaten, Regierungen und Firmen, sondern auch an Institutionen und Privatpersonen.
Folglich müssen Schüler*innen sensibilisiert werden die multidimensional-komplexen Probleme der Zukunft zu visualisieren, um diese nachhaltig lösen zu können und in einer global-vernetzten Welt ihren Platz zu finden. Dieser Aufgabe muss sich Schule als Institution stellen und neben Fachwissen der oben erwähnten nachhaltigen Entwicklung einen wichtigen Stellenwert einräumen. Ziele, wie etwa „*Geschlechtergerechtigkeit*“, „*Hochwertige Bildung*“, „*Weniger Ungleichheiten*“ und „*Frieden*, *Gerechtigkeit und starke Institutionen*“[2] sollten immer im Zentrum eines guten Unterrichts stehen und das Fundament jeder Schule bilden, um jede*n Schüler*in gleichsam in den Blick zu nehmen. Einzelne Ziele jedoch können in verschiedenen Unterrichtsreihen entsprechend besonders in den Fokus genommen und vertieft werden.
„*Bezahlbare und saubere Energie*“, „*Nachhaltige Städte und Gemeinden*/*nachhaltiger Konsum und Produktion*“ und natürlich „*Maßnahmen zum Klimaschutz*“ stellen seit jeher einen genuin-integralen Teil des erfolgreichen und verantwortungsvollen Informatikunterrichts dar. Durch die Schüler*innenarbeiten kann auch hier eine innovativ-nachhaltige Basis als Teil des Unterrichts geschaffen werden, die die Realisierung der 17 Ziele vorantreibt und den Schüler*innen eine nachhaltige Perspektive alternativer, ökologisch verträglicher Energieversorgung vermittelt.
Um diese Entwicklung der 17 Ziele in der Lehrplanarbeit und somit dem Unterricht auch angemessen zu visualisieren, finden sich im Anschluss die 17 Ziele in chronologischer Auflistung, sowie die Farbe, in der diese im Lehrplan zu finden sind:

17 Ziele:

1. Keine Armut
2. Kein Hunger
3. Gesundheit und Wohlergehen
4. Hochwertige Bildung
5. Geschlechtergleichheit
6. Sauberes Wasser und Sanitäreinrichtungen
7. Bezahlbare und Saubere Energie
8. Menschenwürdige Arbeit und Wirtschaftswachstum
9. Industrie, Innovation und Infrastruktur
10. Weniger Ungleichheiten
11. Nachhaltige Städte und Gemeinden
12. Nachhaltiger Konsum und Produktion
13. Maßnahmen zum Klimaschutz
14. Leben unter Wasser
15. Leben an Land
16. Frieden, Gerechtigkeit und starke Institutionen
17. Partnerschaften zur Erreichung der Ziele

2.1 Unterrichtsvorhaben

Die Darstellung der Unterrichtsvorhaben im schulinternen Lehrplan besitzt den Anspruch, sämtliche im Kernlehrplan angeführten Kompetenzen abzudecken. Dies entspricht der Verpflichtung jeder Lehrkraft, Schüler*innen Lerngelegenheiten zu ermöglichen, so dass alle Kompetenzerwartungen des Kernlehrplans von ihnen erfüllt werden können. Die entsprechende Umsetzung erfolgt auf zwei Ebenen: der Übersichts- und der Konkretisierungsebene.

Im „Übersichtsraster Unterrichtsvorhaben“ (Kapitel 2.1.1) wird die für alle Lehrer*innen gemäß Fachkonferenzbeschluss verbindliche Verteilung der Unterrichtsvorhaben dargestellt. Das Übersichtsraster dient dazu, den Kolleg*innen einen schnellen Überblick über die Zuordnung der Unterrichtsvorhaben zu den einzelnen Jahrgangsstufen sowie den im Kernlehrplan genannten Kompetenzen, Inhaltsfeldern und inhaltlichen Schwerpunkten zu verschaffen. Der ausgewiesene Zeitbedarf versteht sich als grobe Orientierungsgröße, die nach Bedarf über- oder unterschritten werden kann.

Um Freiraum für Vertiefungen, besondere Schülerinteressen, aktuelle Themen bzw. die Erfordernisse anderer besonderer Ereignisse (z.B. Praktika, Kursfahrten o.ä.) zu erhalten, wurden im Rahmen dieses schulinternen Lehrplans ca. 75 Prozent der Bruttounterrichtszeit verplant.

Während der Fachkonferenzbeschluss zum „Übersichtsraster Unterrichtsvorhaben“ zur Gewährleistung vergleichbarer Standards sowie zur Absicherung von Lerngruppenübertritten und Lehrkraftwechseln für alle Mitglieder der Fachkonferenz Bindekraft entfalten soll,

beinhaltet die Ausweisung „konkretisierter Unterrichtsvorhaben“ (Kapitel 2.1.2) Beispiele und Materialien, die empfehlenden Charakter haben. Referendar*innen sowie neuen Kolleg*innen dienen diese vor allem zur standardbezogenen Orientierung in der neuen Schule. Da in den folgenden Unterrichtsvorhaben Inhalte in der Regel anhand von Problemstellungen in Anwendungskontexten bearbeitet werden, werden in einigen Unterrichtsvorhaben jeweils mehrere Inhaltsfelder angesprochen.

2.1.1 Übersichtsraster Unterrichtsvorhaben

I) Einführungsphase

Einführungsphase	
Unterrichtsvorhaben E - I **Thema**: Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten Zentrale Kompetenzen: • Argumentieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren Inhaltsfelder: • Informatiksysteme • Informatik, Mensch und Gesellschaft Inhaltliche Schwerpunkte: • Einzelrechner • Dateisystem • Internet • Einsatz von Informatiksystemen **Zeitbedarf**: 8 Stunden	Unterrichtsvorhaben E - II **Thema**: Grundlagen der objektorientierten Analyse, Modellierung und Implementierung anhand von einfachen Beispielkontexten Zentrale Kompetenzen: • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren Inhaltsfelder: • Daten und ihre Strukturierung • Formale Sprachen und Automaten Inhaltliche Schwerpunkte: • Objekte und Klassen • Syntax und Semantik einer Programmiersprache **Zeitbedarf**: 18 Stunden

Einführungsphase	
Unterrichtsvorhaben E - III **Thema**: Objektorientierte Analyse, Modellierung und Implementation anhand von komplexen Beispielkontexten Zentrale Kompetenzen: • Argumentieren	Unterrichtsvorhaben EF – IV **Thema:** Such- und Sortieralgorithmen anhand kontextbezogener Beispiele Zentrale Kompetenzen: • Argumentieren • Modellieren

• Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren Inhaltsfelder: • Algorithmen • Daten und ihre Strukturierung • Formale Sprachen und Automaten Inhaltliche Schwerpunkte: • Objekte und Klassen • Syntax und Semantik einer Programmiersprache • Analyse, Entwurf und Implementierung einfacher Algorithmen **Zeitbedarf**: 24 Stunden	• Darstellen und Interpretieren • Kommunizieren und Kooperieren Inhaltsfelder: • Daten und ihre Strukturierung • Algorithmen Inhaltliche Schwerpunkte: • Algorithmen zum Suchen und Sortieren • Analyse, Entwurf und Implementierung einfacher Algorithmen **Zeitbedarf:** ca. 12 Stunden

II) Qualifikationsphase 1

Qualifikationsphase 1	
Unterrichtsvorhaben Q1 - I **Thema**: Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung Zentrale Kompetenzen: • Argumentieren • Modellieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren Inhaltsfelder: • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten • Informatiksysteme Inhaltliche Schwerpunkte: • Objekte und Klassen • Analyse, Entwurf und Implementierung einfacher Algorithmen • Syntax und Semantik einer Programmiersprache • Nutzung von Informatiksystemen **Zeitbedarf**: ca. 9 Stunden	Unterrichtsvorhaben Q1 - II **Thema**: Rekursive Programmierung an mathematischen und grafischen Beispielen Zentrale Kompetenzen: • Argumentieren • Modellieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren Inhaltsfelder: • Algorithmen • Formale Sprachen und Automaten Inhaltliche Schwerpunkte: • Analyse, Entwurf und Implementierung einfacher Algorithmen • Syntax und Semantik einer Programmiersprache **Zeitbedarf**: ca. 9 Stunden

Qualifikationsphase 1	
Unterrichtsvorhaben Q1 - III **Thema**: Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen Zentrale Kompetenzen: • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren Inhaltsfelder: • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten Inhaltliche Schwerpunkte: • Objekte und Klassen • Analyse, Entwurf und Implementierung einfacher Algorithmen • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache **Zeitbedarf**: ca. 15 Stunden	Unterrichtsvorhaben Q1 - IV **Thema**: Suchen und Sortieren auf linearen Datenstrukturen Zentrale Kompetenzen: • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren Inhaltsfelder: • Algorithmen • Formale Sprachen und Automaten Inhaltliche Schwerpunkte: • Analyse, Entwurf und Implementierung einfacher Algorithmen • Syntax und Semantik einer Programmiersprache **Zeitbedarf**: ca. 15tunden

III) Qualifikationsphase 2

Qualifikationsphase 2	
Unterrichtsvorhaben Q2 - I **Thema**: Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen Zentrale Kompetenzen: • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren Inhaltsfelder: • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten Inhaltliche Schwerpunkte: • Objekte und Klassen • Analyse, Entwurf und Implementierung einfacher Algorithmen • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache **Zeitbedarf**: ca. 24 Stunden	Unterrichtsvorhaben Q2 - II **Thema**: Endliche Automaten und formale Sprachen Zentrale Kompetenzen: • Argumentieren • Modellieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren Inhaltsfelder: • Endliche Automaten und formale Sprachen Inhaltliche Schwerpunkte: • Endliche Automaten • Grammatiken regulärer Sprachen • Möglichkeiten und Grenzen von Automaten und formalen Sprachen **Zeitbedarf**: ca. 20 Stunden

Qualifikationsphase 2	
Unterrichtsvorhaben Q2 - III **Thema**: Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit Zentrale Kompetenzen: • Argumentieren • Kommunizieren und Kooperieren Inhaltsfelder: • Informatiksysteme • Informatik, Mensch und Gesellschaft Inhaltliche Schwerpunkte: • Einzelrechner und Rechnernetzwerke • Grenzen der Automatisierung **Zeitbedarf**: ca. 12 Stunden	

2.1.2 Konkretisierte Unterrichtsvorhaben

I) Einführungsphase

Unterrichtsvorhaben EF - I

Thema: Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten

17 Ziele:

Gesundheit und Wohlergehen
Hochwertige Bildung
Geschlechtergleichheit
Bezahlbare und Saubere Energie
Menschenwürdige Arbeit und Wirtschaftswachstum
Industrie, Innovation und Infrastruktur
Nachhaltiger Konsum und Produktion
Maßnahmen zum Klimaschutz
Frieden, Gerechtigkeit und starke Institutionen

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
1. Information, deren Kodierung und Speicherung a) Informatik als Wissenschaft der Ver-	**Die Schüler*innen**

arbeitung von Informationen b) Darstellung von Informationen in Schrift, Bild und Ton c) Speichern von Daten mit informatischen Systemen am Beispiel der Schulrechner d) Vereinbarung von Richtlinien zur Datenspeicherung auf den Schulrechnern (z.B. Ordnerstruktur, Dateibezeichner usw.)	• beschreiben und erläutern den Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A) • nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D) • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K)
2. Informations- und Datenübermittlung in Netzen a) „Sender-Empfänger-Modell“ und seine Bedeutung für die Eindeutigkeit von Kommunikation b) Informatische Kommunikation in Rechnernetzen am Beispiel der Schulnetzwerks (z.B. Benutzeranmeldung, Netzwerkordner, Zugriffsrechte) c) Grundlagen der technischen Umsetzung von Rechnerkommunikation am Beispiel des Internets (z.B. Netzwerkadresse, Paketvermittlung, Protokoll) d) Richtlinien zum verantwortungsvollen Umgang mit dem Internet	
3. Aufbau informatischer Systeme a) Identifikation typischer Komponenten informatischer Systeme und anschließende Beschränkung auf das Wesentliche b) Identifikation des EVA-Prinzips (Eingabe-Verarbeitung-Ausgabe) als Prinzip der Verarbeitung von Daten und Grundlage der „Von – Neumann - Architektur“	

Unterrichtsvorhaben EF - II

Thema: Grundlagen der objektorientierten Analyse, Modellierung und Implementierung anhand von einfachen Beispielkontexten

17 Ziele:

Gesundheit und Wohlergehen
Hochwertige Bildung
Geschlechtergleichheit
Bezahlbare und Saubere Energie
Menschenwürdige Arbeit und Wirtschaftswachstum
Industrie, Innovation und Infrastruktur
Nachhaltiger Konsum und Produktion
Maßnahmen zum Klimaschutz
Frieden, Gerechtigkeit und starke Institutionen

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
1. Identifikation von Objekten a) An einem einfachen Beispiel werden Objekte im Sinne der objektorientierten Modellierung eingeführt b) Manche Objekte sind typgleich und werden so zu einer Objektsorte bzw. Klasse zusammengefasst. c) Klassen werden in Diagrammen visualisiert und mit sinnvollen Attributen und Methoden versehen d) Vertiefung: Modellierung weiterer Beispiele ähnlichen Musters	**Die Schüler*innen** • ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen(M) • modellieren Klassen mit ihren Attributen und ihren Methoden (M) • stellen Attribute und Methoden in einem Klassendiagramm dar (D) • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen zu (M) • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I) • interpretieren Fehlermeldungen und korrigieren den Quellcode • stellen den Zustand eines Objektes dar (D) • implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), • analysieren und erläutern einfache Algorithmen und Programme (A)
2. Einfache Greenfoot – Szenarien a) Analyse von vorgegebenen Klassen b) Grundlagen der Vererbung c) Abfrage von Zuständen	

Unterrichtsvorhaben EF - III

Thema: Objektorientierte Analyse, Modellierung und Implementation anhand von komplexen Beispielkontexten

17 Ziele:

Gesundheit und Wohlergehen
Hochwertige Bildung
Geschlechtergleichheit
Bezahlbare und Saubere Energie
Menschenwürdige Arbeit und Wirtschaftswachstum
Industrie, Innovation und Infrastruktur
Nachhaltiger Konsum und Produktion
Maßnahmen zum Klimaschutz
Leben unter Wasser
Leben an Land
Frieden, Gerechtigkeit und starke Institutionen

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
1. Entwicklung von Spielen oder Simulationen mit Interaktion zwischen unterschiedlichen Objekten in Greenfoot a) Schrittweise Implementation der Projekte b) Kontinuierliches Bewegen eines Objektes mit der while – Schleife oder do – while – Schleife c) Dokumentation der Klassen des Projekts d) Kollisionsabfragen e) Verwendung von Objektreferenzen	**Die Schüler*innen** • analysieren und erläutern eine objektorientierte Modellierung (A) • stellen die Kommunikation zwischen Objekten grafisch dar (M) • ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M) • modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M) • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M) • ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M) • modellieren Klassen unter Verwendung von Vererbung (M) • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I) • testen Programme schrittweise an-
2. Die Klasse String a) Primitiver Datentyp char b) Strings als Reihen von chars c) String – Methoden d) Operationen auf Strings	
3. Vertiefung der Vererbung a) Analyse und Erläuterung einer Basisversion einer Klasse b) Realisierung von Erweiterungen zur Basisklasse mit Vererbung c) Verallgemeinerung und Reflexion des Prinzips der Vererbung am Beispiel der Spezialisierung d) Überschreiben von Methoden	

	hand von Beispielen (I) • modifizieren einfache Algorithmen und Programme (I) • stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D) • dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D).

Unterrichtsvorhaben EF - IV

Thema: Such- und Sortieralgorithmen anhand kontextbezogener Beispiele

17 Ziele:

Gesundheit und Wohlergehen
Hochwertige Bildung
Geschlechtergleichheit
Bezahlbare und Saubere Energie
Menschenwürdige Arbeit und Wirtschaftswachstum
Industrie, Innovation und Infrastruktur
Nachhaltiger Konsum und Produktion
Maßnahmen zum Klimaschutz
Frieden, Gerechtigkeit und starke Institutionen

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
1. Explorative Erarbeitung eines Sortierverfahrens a) Sortierprobleme im Kontext informatorischer Systeme und im Alltag (z. B. Datensortierung, Tabellenkalkulation, Telefonbuch usw.) b) Vergleich zweier Elemente als Grundlage eines Sortieralgorithmus c) Erarbeitung eines Sortieralgorithmus durch die Schüler*innen	**Die Schüler*innen** • beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeit und Speicherplatzbedarf (A) • entwerfen einen weiteren Algorithmus zu Sortieren (M) • analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D)
2. Systematisierung von Algorithmen und Effizienzbetrachtungen a) Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode (auf jeden Fall: Sortieren durch Vertauschen, Sortieren durch Auswählen) b) Anwendung von Sortieralgorithmen	

auf verschiedene Beispiele c) Bewertung von Algorithmen anhand der nötigen Vergleiche d) Variante des Sortierens durch Auswählen (Nutzung eines einzigen oder zweier Felder bzw. lediglich eines zusätzlichen Ablageplatzes oder mehrerer neuer Ablageplätze) e) Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des Speicherplatzbedarfs	
3. Binäre Suche auf sortierten Daten a) Suchaufgaben im Alltag und im Kontext informatischer Systeme b) Evtl. Simulationsspiel zum effizienten Suchen mit binärer Suche c) Effizienzbetrachtung zur binären Suche	

II) Qualifikationsphase
Unterrichtsvorhaben Q1 – 1
Thema: Wiederholung der objektorientierten Modellierung und Programmierung

17 Ziele:

Hochwertige Bildung
Geschlechtergleichheit
Bezahlbare und Saubere Energie
Menschenwürdige Arbeit und Wirtschaftswachstum
Industrie, Innovation und Infrastruktur
Nachhaltiger Konsum und Produktion
Maßnahmen zum Klimaschutz
Frieden, Gerechtigkeit und starke Institutionen
Partnerschaften zur Erreichung der Ziele

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
Wiederholung und Erweiterung der objektorientierten Modellierung und Programmierung durch Analyse und Erweiterung eines kontextbezogenen Beispiels a) Analyse der Problemstellung	Die Schüler*innen • analysieren und erläutern objektorientierte Modellierungen (A) • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A) • modellieren Klassen mit ihren Attribu-

b) Analyse der Modellierung (Implementationsdiagramm) c) Erweiterung der Modellierung im Implementationsdiagramm (Vererbung, abstrakte Klassen) d) Kommunikation zwischen mindestens zwei Objekten(grafische Darstellung) e) Dokumentation von Klassen f) Implementierung der Anwendung oder Teilen Anwendung	ten, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M) • ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M) • modellieren abstrakte und nicht abstrakte Klassen unter Vererbung durch Spezialisieren und Generalisieren (M) • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I) • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I) • wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I) • interpretieren Fehlermeldungen und korrigieren den Quellcode (I) • stellen Klassen und ihre Beziehungen in Diagrammen dar (D) • stellen die Kommunikation zwischen Objekten grafisch dar (D)

Unterrichtsvorhaben Q1 – 2
Thema: Rekursive Programmierung an mathematischen und grafischen Beispielen

17 Ziele:

Hochwertige Bildung
Geschlechtergleichheit
Bezahlbare und Saubere Energie
Menschenwürdige Arbeit und Wirtschaftswachstum
Industrie, Innovation und Infrastruktur
Nachhaltiger Konsum und Produktion
Maßnahmen zum Klimaschutz
Partnerschaften zur Erreichung der Ziele

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
Rekursive und iterative Berechnung von mathematischen Folgen a) Analyse und Berechnung mathematischer Folgen b) Implementation rekursiver Methoden zur Berechnung von Folgen c) Analyse von rekursiven Methoden unter Laufzeit- und Speicherplatzaspekten d) Analyse einfacher Lindenmayer – Systeme e) Implementierung einfacher Fraktale mit Hilfe einer Turtle – Grafikszene f) Abbruchbedingungen für die Methoden und Aussagen über das Laufzeitverhalten	Die Schüler*innen • analysieren und erläutern Algorithmen und Programme (A) • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D) • entwickeln iterative und rekursive Algorithmen unter der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M) • implementieren iterative und rekursive Algorithmen (I) • beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A)

Unterrichtsvorhaben Q1 – 3
Thema: Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen

17 Ziele:

Gesundheit und Wohlergehen
Hochwertige Bildung
Geschlechtergleichheit
Bezahlbare und Saubere Energie
Menschenwürdige Arbeit und Wirtschaftswachstum
Industrie, Innovation und Infrastruktur
Nachhaltiger Konsum und Produktion
Maßnahmen zum Klimaschutz
Frieden, Gerechtigkeit und starke Institutionen

Unterrichtssequenzen	Zu entwickelnde Kompetenzen

<table>
<tr><td>1. Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse Queue
a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen
b) Erarbeitung der Funktionalität der Klasse Queue
c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Queue
d) Erstellen eines Implementierungsdiagramms für die Anwendung</td><td rowspan="3">Die Schüler*innen
• erläutern Operationen dynamischer (linearer oder nicht linearer) Datenstrukturen (A)
• analysieren und erläutern Algorithmen und Programme (A)
• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A)
• entwickeln iterative und rekursive Algorithmen unter der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M)
• ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M)
• implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I)
• beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A)
• nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I)
• interpretieren Fehlermeldungen und korrigieren den Quellcode (I)
• testen Programme systematisch anhand von Beispielen (I)
• stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D)</td></tr>
<tr><td>2. Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse Queue
a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen
b) Erarbeitung der Funktionalität der Klasse Queue
c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Queue
d) Erstellen eines Implementationsdiagramm für die Anwendung</td></tr>
<tr><td>3. Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse List
a) Erarbeitung der Vorteile der Klasse List im Gegensatz zu den bereits bekannten linearen Strukturen
b) Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse List</td></tr>
</table>

Unterrichtsvorhaben Q1 – 4

Thema: Suchen und Sortieren auf linearen Datenstrukturen

17 Ziele:

Gesundheit und Wohlergehen
Hochwertige Bildung
Innovation und Infrastruktur
Nachhaltiger Konsum und Produktion
Frieden, Gerechtigkeit und starke Institutionen
Partnerschaften zur Erreichung der Ziele

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
1. Suchen von Dateien in Listen und Arrays a) Lineare Suche in Listen und Arrays b) Binäre Suche in Arrays als Beispiel für rekursives Problemlösen c) Untersuchung der beiden Suchverfahren hinsichtlich ihrer Effizienz (Laufzeitverhalten, Speicherbedarf) **2. Sortieren in Listen und Arrays – Entwicklung und Implementierung von iterativen und rekursiven Sortierverfahren** a) Entwicklung und Implementierung eines einfachen Sortierverfahrens für eine Liste b) Implementierung eines einfachen Sortierverfahrens für ein Feld c) Entwicklung eines rekursiven Sortierverfahrens für ein Feld (z. B. Sortieren durch Mischen)	Die Schüler*innen • analysieren und erläutern Algorithmen und Programme (A) • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A) • beurteilen die Effizienz von Algorithmen unter der Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A) • entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ • modifizierte Algorithmen und Programme (I) • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I) • implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I) • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I) • interpretieren Fehlermeldungen und korrigieren den Quellcode • testen Programme systematisch anhand von Beispielen (I)
3. Untersuchung der Effizienz der Sortierverfahren „Sortieren durch direktes Einfügen“ und „Quicksort“ auf linearen Listen a) Grafische Veranschaulichung der Sortierverfahren b) Untersuchung der Anzahl der Vergleichsoperationen und des Speicherbedarfs bei beiden Sortier-	

verfahren c) Beurteilung der Effizienz der beiden Sortierverfahren	• stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D)

Unterrichtsvorhaben Q2 – 1

Thema: Modellierung und Implementierung von Anwendungen mit dynamischen nichtlinearen Datenstrukturen

17 Ziele:

Gesundheit und Wohlergehen
Hochwertige Bildung
Innovation und Infrastruktur
Nachhaltiger Konsum und Produktion
Frieden, Gerechtigkeit und starke Institutionen
Partnerschaften zur Erreichung der Ziele

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
1. Analyse von Baumstrukturen in verschiedenen Kontexten a) Grundlegende Begriffe (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit) b) Aufbau und Darstellung von binären Bäumen anhand von Baumstrukturen in verschiedenen Kontexten	Die Schüler*innen • erläutern Operationen dynamischer (linearer oder nicht – linearer) Datenstrukturen (A) • analysieren und erläutern Algorithmen und Programme (A) • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A) • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M) • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M) • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M) • verwenden bei der Modellierung geeigneter Problemstellungen die Mög-
2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse BinäryTree a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext b) Modellierung eines Entwurf Diagramms und Entwicklung eines Implementationsdiagramms c) Erarbeitung der Klasse BinäryTree und beispielhafte Anwendung der Operationen d) Implementierung der Anwendung oder Teilen der Anwendung	

e) e) Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf	lichkeiten der Polymorphie (M) • entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M) • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I) • modifizieren Algorithmen und Programme (I) • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I) • interpretieren und korrigieren den Quellcode (I) • testen Programme systematisch anhand von Beispielen (I) • stellen lineare und nichtlineare Strukturen grafisch dar unter erläutern ihren Aufbau (D) • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D)
3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse BinäryTree a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen b) Modellierung eines Entwurf Diagramms und Entwicklung eines Implementationsdiagramms, grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften c) Erarbeitung der Klasse BinäryTree und Einführung des Interface Comparable – Content zur Realisierung einer geeigneten Ordnungsrelation d) Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums	

Unterrichtsvorhaben Q2 – 2

Thema: Endliche Automaten und formale Sprachen

17 Ziele:

Gesundheit und Wohlergehen
Hochwertige Bildung
Geschlechtergleichheit
Bezahlbare und Saubere Energie
Menschenwürdige Arbeit und Wirtschaftswachstum
Industrie, Innovation und Infrastruktur
Nachhaltiger Konsum und Produktion
Maßnahmen zum Klimaschutz
Frieden, Gerechtigkeit und starke Institutionen

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
1. Endliche Automaten	Die Schüler*innen

a) Vom Automaten in den Schüler*innen bekannten Kontexten zur formalen Beschreibung eines endlichen Automaten b) Untersuchung, Modifikation und Entwicklung regulärer Grammatiken c) Entwicklung von endlichen Automaten zum Erkennen regulärer Sprachen die durch Grammatiken gegeben werden d) Entwicklung regulärer Grammatiken zu endlichen Automaten **2. Untersuchung und Entwicklung von Grammatiken regulärer Sprachen** a) Erarbeitung der formalen Darstellung regulärer Grammatiken b) Untersuchung, Modifikation und Entwicklung von Grammatiken c) Entwicklung von endlichen Automaten zum Erkennen regulärer Sprachen die durch Grammatiken gegeben werden d) Entwicklung regulärer Grammatiken zu endlichen Automaten	• analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens auf bestimmte Eingaben (A) • analysieren und erläutern Grammatiken regulärer Sprachen (A) • zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (A) • ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A) • entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M) • entwickeln zur akzeptierten Sprache eines Automaten die zugehörige F Grammatik (M) • entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M) • modifizieren Grammatiken regulärer Sprachen (M) • entwickeln zu einer regulären Sprachen eine Grammatik, die die Sprache erzeugt (M) • stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D) • ermitteln die Sprache, die ein endlicher Automat erfüllt (D) • beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D)

Unterrichtsvorhaben Q2 – 3

Thema: Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

17 Ziele:

Gesundheit und Wohlergehen
Hochwertige Bildung
Innovation und Infrastruktur
Nachhaltiger Konsum und Produktion
Leben unter Wasser
Leben an Land
Frieden, Gerechtigkeit und starke Institutionen
Partnerschaften zur Erreichung der Ziele

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
1. Von Neumann – Architektur und die Ausführung maschinennaher Programme a) Prinzipieller Aufbau einer von Neumann – Architektur mit CPU, Rechenwerk, Register und Hauptspeicher b) Einige maschinennahe Befehle und ihre Repräsentationen in einem Binär – Code, der in einem Register gespeichert werden kann c) Analyse und Erläuterung der Funktionsweise eines einfachen maschinennahen Programms **2. Grenzen der Automatisierbarkeit** a) Vorstellung des Halteproblems b) Unlösbarkeit des Halteproblems c) Beurteilung des Einsatzes von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen	Die Schüler*innen • erläutern die Ausführung eines einfachen maschinennahen Programms sowie die Datenspeicherung auf einer „Von – Neumann – Architektur (A) • untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A)

2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

In Absprache mit der Lehrerkonferenz sowie unter Berücksichtigung des Schulprogramms hat die Fachkonferenz Informatik des Gesamtschule Kaarst die folgenden fachmethodischen und fachdidaktischen Grundsätze beschlossen. In diesem Zusammenhang beziehen sich die Grundsätze 1 bis 14 auf fächerübergreifende Aspekte, die auch Gegenstand der Qualitätsanalyse sind, die Grundsätze 15 bis 21 sind fachspezifisch angelegt.

Überfachliche Grundsätze:

1. Geeignete Problemstellungen zeichnen die Ziele des Unterrichts vor und bestimmen die Struktur der Lernprozesse.
2. Inhalt und Anforderungsniveau des Unterrichts entsprechen dem Leistungsvermögen der Schüler/innen.
3. Die Unterrichtsgestaltung ist auf die Ziele und Inhalte abgestimmt.
4. Medien und Arbeitsmittel sind schülernah gewählt.
5. Die Schüler/innen erreichen einen Lernzuwachs.
6. Der Unterricht fördert eine aktive Teilnahme der Schüler/innen.
7. Der Unterricht fördert die Zusammenarbeit zwischen den Schülern/innen und bietet ihnen Möglichkeiten zu eigenen Lösungen.
8. Der Unterricht berücksichtigt die individuellen Lernwege der einzelnen Schüler/innen.
9. Die Schüler/innen erhalten Gelegenheit zu selbstständiger Arbeit und werden dabei unterstützt.
10. Der Unterricht fördert strukturierte und funktionale Partner- bzw. Gruppenarbeit.
11. Der Unterricht fördert strukturierte und funktionale Arbeit im Plenum.
12. Die Lernumgebung ist vorbereitet; der Ordnungsrahmen wird eingehalten.
13. Die Lehr- und Lernzeit wird intensiv für Unterrichtszwecke genutzt.
14. Es herrscht ein positives pädagogisches Klima im Unterricht.

Fachliche Grundsätze:

15. Der Unterricht unterliegt der Wissenschaftsorientierung und ist dementsprechend eng verzahnt mit seiner Bezugswissenschaft.
16. Der Unterricht ist problemorientiert und soll von realen Problemen ausgehen und sich auf solche rückbeziehen.

17. Der Unterricht folgt dem Prinzip der Exemplarizität und soll ermöglichen, informatische Strukturen und Gesetzmäßigkeiten in den ausgewählten Problemen und Projekten zu erkennen.
18. Der Unterricht ist anschaulich sowie gegenwarts- und zukunftsorientiert und gewinnt dadurch für die Schülerinnen und Schüler an Bedeutsamkeit.
19. Der Unterricht ist handlungsorientiert, d.h. projekt- und produktorientiert angelegt.
20. Im Unterricht werden sowohl für die Schule didaktisch reduzierte als auch reale Informatiksysteme aus der Wissenschafts-, Berufs- und Lebenswelt eingesetzt.
21. Der Unterricht beinhaltet reale Begegnung mit Informatiksystemen.

2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

Auf der Grundlage von §13 - §16 der APO-GOSt sowie Kapitel 3 des Kernlehrplans Informatik für die gymnasiale Oberstufe hat die Fachkonferenz des Gesamtschule Kaarst im Einklang mit dem entsprechenden schulbezogenen Konzept die nachfolgenden Grundsätze zur Leistungsbewertung und Leistungsrückmeldung beschlossen. Die nachfolgenden Absprachen stellen die Minimalanforderungen an das lerngruppenübergreifende gemeinsame Handeln der Fachgruppenmitglieder dar. Bezogen auf die einzelne Lerngruppe kommen ergänzend weitere der in den Folgeabschnitten genannten Instrumente der Leistungsüberprüfung zum Einsatz.

2.3.1 Beurteilungsbereich Klausuren

Verbindliche Absprachen:

Bei der Formulierung von Aufgaben werden die für die Abiturprüfungen geltenden Operatoren des Faches Informatik schrittweise eingeführt, erläutert und dann im Rahmen der Aufgabenstellungen für die Klausuren benutzt.

Instrumente:

Einführungsphase:	1 Klausur je Halbjahr
Dauer der Klausur:	2 Unterrichtsstunden
Grund- und Leistungskurse Q 1:	2 Klausuren je Halbjahr
Dauer der Klausuren:	2 Unterrichtsstunden (Grundkurs), 3 (Leistungskurs)

Grund- und Leistungskurse Q 2.1:	2 Klausuren
Dauer der Klausuren:	3 Unterrichtsstunden (Grundkurs), 4 (Leistungskurs)
Grund- und Leistungskurse Q 2.2:	1 Klausur unter Abiturbedingungen

Anstelle einer Klausur kann gemäß dem Beschluss der Lehrerkonferenz in Q 1.2 eine Facharbeit geschrieben werden.
Die Aufgabentypen, sowie die Anforderungsbereiche I-III sind entsprechend den Vorgaben in Kapitel 3 des Kernlehrplans zu beachten.

Kriterien der Bewertung der schriftlichen Arbeit
Die Bewertung der schriftlichen Leistungen in Klausuren erfolgt über ein Raster mit Hilfspunkten, die im Erwartungshorizont den einzelnen Kriterien zugeordnet sind.
Spätestens ab der Qualifikationsphase orientiert sich die Zuordnung der Hilfspunktsumme zu den Notenstufen an dem Zuordnungsschema des Zentralabiturs.
Von diesem kann aber im Einzelfall begründet abgewichen werden, wenn sich z.B. besonders originelle Teillösungen nicht durch Hilfspunkte gemäß den Kriterien des Erwartungshorizontes abbilden lassen oder eine Abwertung wegen besonders schwacher Darstellung (APO-GOSt §13 (2)) angemessen erscheint.
Die Note ausreichend (5 Punkte) soll bei Erreichen von 45 % der Hilfspunkte erteilt werden.

2.3.2 Beurteilungsbereich Sonstige Mitarbeit

Den Schüler*innen werden die Kriterien zum Beurteilungsbereich „sonstige Mitarbeit“ zu Beginn des Schuljahres genannt.

Verbindliche Absprachen der Fachkonferenz

- Alle Schülerinnen und Schüler führen in der Einführungsphase in Kleingruppen ein Kurzprojekt durch und fertigen dazu eine Arbeitsmappe mit Arbeitstagebuch an. Dies wird in die Note für die Sonstige Mitarbeit einbezogen.

- In der Qualifikationsphase erstellen, dokumentieren und präsentieren die Schülerinnen und Schüler in Kleingruppen ein anwendungsbezogenes Softwareprodukt. Dies wird in die Note für die Sonstige Mitarbeit einbezogen.

Leistungsaspekte

Mündliche Leistungen

Beteiligung am Unterrichtsgespräch

Zusammenfassungen zur Vor- und Nachbereitung des Unterrichts

Präsentation von Arbeitsergebnissen

Referate

Mitarbeit in Partner-/Gruppenarbeitsphasen

Praktische Leistungen am Computer

Implementierung, Test und Anwendung von Informatiksystemen

Sonstige schriftliche Leistungen

Arbeitsmappe und Arbeitstagebuch zu einem durchgeführten Unterrichtsvorhaben

Lernerfolgsüberprüfung durch kurze schriftliche Übungen

In Kursen, in denen höchstens 50% der Kursmitglieder eine Klausur schreiben, finden schriftliche Übungen mindestens einmal pro Kurshalbjahr statt, in anderen Kursen entscheidet über die Durchführung die Lehrkraft.

Schriftliche Übung dauern ca. 20 Minuten und umfassen den Stoff der letzten ca. 4–6 Stunden.

Bearbeitung von schriftlichen Aufgaben im Unterricht

Kriterien

Die folgenden allgemeinen Kriterien gelten sowohl für die mündlichen als auch für die schriftlichen Formen der sonstigen Mitarbeit.

Die Bewertungskriterien stützen sich auf

- die Qualität der Beiträge,
- die Quantität der Beiträge und
- die Kontinuität der Beiträge.

Besonderes Augenmerk ist dabei auf

- die sachliche Richtigkeit,
- die angemessene Verwendung der Fachsprache,
- die Darstellungskompetenz,
- die Komplexität und den Grad der Abstraktion,
- die Selbstständigkeit im Arbeitsprozess,
- die Präzision und
- die Differenziertheit der Reflexion zu legen.

Bei Gruppenarbeiten auch auf

- das Einbringen in die Arbeit der Gruppe,
- die Durchführung fachlicher Arbeitsanteile und
- die Qualität des entwickelten Produktes.

Bei Projektarbeit darüber hinaus auf

- die Dokumentation des Arbeitsprozesses,
- den Grad der Selbstständigkeit,
- die Reflexion des eigenen Handelns und
- die Aufnahme von Beratung durch die Lehrkraft

2.3.3 Grundsätze der Leistungsrückmeldung und Beratung

Die Grundsätze der Leistungsbewertung werden zu Beginn eines jeden Halbjahres den Schüler*innen transparent gemacht. Leistungsrückmeldungen können erfolgen

- nach einer mündlichen Überprüfung,
- bei Rückgabe von schriftlichen Leistungsüberprüfungen,
- nach Abschluss eines Projektes,
- nach einem Vortrag oder einer Präsentation,
- bei auffälligen Leistungsveränderungen,
- auf Anfrage,
- als Quartalsfeedback und
- zu Eltern- oder Schülersprechtagen.

Die Leistungsrückmeldung kann

- durch ein Gespräch mit der Schülerin oder dem Schüler,
- durch einen Feedbackbogen,
- durch die schriftliche Begründung einer Note oder
- durch eine individuelle Lern-/Förderempfehlung

  erfolgen.

2.4 Lehr- und Lernmittel

Die Fachkonferenz hat sich entschieden zum Einstieg in das Fach Informatik in der Sek II das Lehrwerk „Informatik 1“ von Schöningh zu verwenden. Das Werk beinhaltet die Unterrichtsinhalte der EF und der Q1. Des Weiteren werden Internetseiten für Computer Grundlagen und zur Einführung und Festigung der Objektorientierten Modellierung an Beispielkontexten verwendet. In der EF werden die Seiten www.easy4me.info, Alois Klotz, Dienstleistungen in der automatischen Datenverarbeitung und Informationstechnik und www.informatikzentrale.de, verantwortlich Berthold Metz, Freiburg, Einführung in die Objektorientierte Modellierung, verwendet. Dazu wird die interaktive Java – Entwicklungsumgebung Greenfoot kostenfrei aus dem Internet runtergeladen. Greenfoot wurde für Ausbildungszwecke entwickelt und erlaubt die Programmierung einfacher zweidimensionaler Anwendungen.

In der Q1 wird die Internetseite www.u-helmich.de, verantwortlich Ulrich Helmich, Rahden, verwendet. Herr Helmich veröffentlicht Unterrichtsmaterialien zur Fortführung der Objektorientierten Modellierung mit der kostenfreien Programmierumgebung BlueJ. In BlueJ wird den Schüler*innen die Programmierung mit Java in einer optimalen Umgebung nähergebracht.

Der Unterricht ist gemäß der Zusammenstellung der Unterrichtsvorhaben durch weitere Materialien zu ergänzen. Dieses Unterrichtsmaterial wird die Fachschaft Informatik im Rahmen des weiteren Ausbaus der Oberstufe beschaffen. Dabei orientiert sich die Fachschaft am LEHRPLANNAVIGATOR des Faches Informatik, um über einen Überblick über die zugelassenen Lernmittel zu verfügen.

3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen

Die Fachkonferenz Informatik hat sich im Rahmen des Schulprogramms für folgende zentrale Schwerpunkte entschieden:

Zusammenarbeit mit anderen Fächern

Im Informatikunterricht werden Kompetenzen anhand informatischer Inhalte in verschiedenen Anwendungskontexten erworben, in denen Schülerinnen und Schülern aus anderen Fächern Kenntnisse mitbringen können. Diese können insbesondere bei der Auswahl und

Bearbeitung von Softwareprojekten berücksichtigt werden und in einem hinsichtlich der informatischen Problemstellung angemessenem Maß in den Unterricht Eingang finden. Da im Inhaltsfeld Informatik, Mensch und Gesellschaft auch gesellschaftliche und ethische Fragen im Unterricht angesprochen werden, soll eine mögliche Zusammenarbeit mit den Fächern Sozialwissenschaften und Philosophie in einer gemeinsamen Fachkonferenz ausgelotet werden.

Projekttage

Alle zwei Jahre werden am Gesamtschule Kaarst Projekttage angeboten. Die Fachkonferenz Informatik bietet in diesem Zusammenhang mindestens ein Projekt für Schülerinnen und Schüler der gymnasialen Oberstufe an.

Vorbereitung auf die Erstellung der Facharbeit

Möglichst schon zweiten Halbjahr der Einführungsphase, spätestens jedoch im ersten Halbjahr des ersten Jahres der Qualifikationsphase werden im Unterricht an geeigneten Stellen Hinweise zur Erstellung von Facharbeiten gegeben. Das betrifft u. a. Themenvorschläge, Hinweise zu den Anforderungen und zur Bewertung. Es wird vereinbart, dass nur Facharbeiten vergeben werden, die mit der eigenständigen Entwicklung eines Softwareproduktes verbunden sind.

Exkursionen

In der Einführungsphase wird im Rahmen des Unterrichtsvorhabens „Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes“ eine Exkursion zum Heinz Nixdorf MuseumsForum durchgeführt. Die außerunterrichtliche Veranstaltung wird im Unterricht vor- und nachbereitet.

4 Qualitätssicherung und Evaluation

Das schulinterne Curriculum stellt keine starre Größe dar, sondern ist als „lebendes Dokument“ zu betrachten. Dementsprechend sind die Inhalte stetig zu überprüfen, um ggf.

Modifikationen vornehmen zu können. Die Fachkonferenz (als professionelle Lerngemeinschaft) trägt durch diesen Prozess zur Qualitätsentwicklung und damit zur Qualitätssicherung des Faches bei.

Durch Diskussion der Aufgabenstellung von Klausuren in Fachdienstbesprechungen und eine regelmäßige Erörterung der Ergebnisse von Leistungsüberprüfungen wird ein hohes Maß an fachlicher Qualitätssicherung erreicht. Das schulinterne Curriculum (siehe 2.1) ist zunächst bis 2022 für den ersten Durchgang durch die gymnasiale Oberstufe nach Erlass des Kernlehrplanes verbindlich. Erstmalig nach Ende der Einführungsphase im Sommer 2020, werden in einer Sitzung der Fachkonferenz Erfahrungen ausgetauscht und ggf. Änderungen für den nächsten Durchgang der Einführungsphase beschlossen, um erkannten ungünstigen Entscheidungen schnellstmöglich entgegenwirken zu können.

Nach Abschluss des Abiturs 2022 wir die Fachkonferenz Informatik auf der Grundlage ihrer Unterrichtserfahrungen eine Gesamtsicht des schulinternen Curriculums vornehmen und ggf. eine Beschlussvorlage für die erste Fachkonferenz des folgenden Schuljahres erstellen.