Schuleigener Arbeitsplan für das Fach Informatik in der Sekundarstufe II

CAG
CLEMENS-AUGUST-GYMNASIUM
EUROPASCHULE

Einführungsphase

Zeit-raum/ Dauer	Lernfeld/ Modul/ ggf. Thema	Lernfeldbezogene Kompetenzen (Die Schüler/ Schülerinnen ….)	Allgemeine Kompe-tenzen[1]	Sonstiges (Werkzeuge, An-merkungen, Bezüge, didak-tisch-methodische Hinweise)
ca. 12 Wochen	Einführung in die (textbasierte) Programmierung; Algorithmen und Datenstrukturen	• benennen Anweisung, Sequenz, Schleife und Verzweigung als Grundbausteine eines Algorithmus. • entwerfen und implementieren Algorithmen unter zielgerichteter Verwendung der elementaren Kontrollstrukturen. • stellen Algorithmen in standardisierter Form dar. • erläutern das Prinzip der Speicherung von Werten in Variablen. • verwenden Variablen und Wertzuweisungen in Algorithmen. • stellen die Belegung von Variablen bei der Ausführung eines Algorithmus in Form einer Trace-Tabelle dar. • verwenden und erstellen Operationen zur strukturierten Implementierung von Algorithmen. • entwerfen und implementieren Algorithmen unter Verwendung elementarer Zeichenkettenoperationen	PK1.1,PK1.2 PK2.1, PK2.2, PK3.1, PK3.2 IK1.2, IK2.1, IK2.2	Grundbegriffe: Algorithmus, Information, Daten z.B. Scratch bzw. code.org und Python (als textbasierte Programmiersprache)
ca. 4 Wochen	Projektarbeit	Projekt zur Festigung der Kompetenzen der Algorithmik und zur Motivation der folgenden Unterrichtsreihen		z.B. Python

[1] Kompetenzen entsprechend der curricularen Vorgaben für Informatik in der gymnasialen Oberstufe, vgl. http://db2.nibis.de/1db/cuvo/datei/inf_go_kc_druck_2017.pdf

Zeit-raum/ Dauer	**Lernfeld/ Modul/ ggf. Thema**	**Lernfeldbezogene Kompetenzen** (Die Schüler/ Schülerinnen ….)	**Allgemeine Kompe-tenzen[1]**	**Sonstiges (Werkzeuge, An-merkungen, Bezüge, didak-tisch-methodische Hinweise)**
ca. 3 Wochen	Einführung in die Kryptologie	• beschreiben das Prinzip der Transposition und der Substitution zur Verschlüsselung von Daten. • implementieren monoalphabetische Verfahren, u. a. Caesar-Verfahren. • erläutern das Prinzip der Häufigkeitsanalyse. • beurteilen die Sicherheit einfacher Verschlüsselungsverfahren.	PK2.2, PK3.3 IK2.3	z.B. Spioncamp-Koffer
ca. 7 Wochen	Einführung in die Codierung Netzwerke Datenschutz	• beschreiben zentrale Komponenten eines Informatiksystems und deren Zusammenspiel. • beschreiben und begründen den dezentralen Aufbau des Internets. • nennen die zentralen Komponenten des Internets, u. a. Client, Server, Router, DNS und erläutern ihre Funktion. • beschreiben die Kommunikationswege im Internet. • beschreiben Aspekte zur Sicherheit der Kommunikation im Internet. • beschreiben grundlegende Codierungen von Daten, u. a. Dualzahlen, ASCII. • erläutern die rechtlichen Rahmenbedingungen für den Umgang mit ihren persönlichen Daten, wie z.B. informationelle Selbstbestimmung und Datenschutzrichtlinien	PK3.3, IK1.1, IK3.1, IK3.3	*z.B. GoldCity & NewTown* Filius
ca. 4 Wochen	Projektarbeit	• entwerfen und implementieren Algorithmen unter zielgerichteter Verwendung der elementaren Kontrollstrukturen.		*z.B. Python*

Qualifikationsphase 1. Halbjahr

Zeit-raum/ Dauer	Lernfeld/ Modul/ ggf. Thema	Lernfeldbezogene Kompetenzen	Allgemeine Kompe-tenzen	Sonstiges (Werkzeuge, An-merkungen, Bezüge, didak-tisch-methodische Hinweise)
ca. 7 Wochen	Datenbanken	• erläutern den Aufbau relationaler Datenbanken unter Verwendung der Begriffe Datensatz, Attribut, Primär-/Fremdschlüssel und Tabelle. • nennen Beispiele für Einfüge-, Änderungs- und Löschanomalien. • untersuchen ein gegebenes Datenbankschema auf Anomalien und Redundanzen. • formulieren einfache Abfragen und Verbundabfragen über mehrere Tabellen. • formulieren Abfragen an Datenbanken unter Verwendung von Aggregatfunktionen. • diskutieren Chancen und Risiken der automatisierten Datenanalyse.	PK1.2, PK1.3 PK2.2, PK3.1 PK3.3 IK2.4, IK4.1 IK4.2	Fitnesscenter-Datenbank; SQL-Tutorial der Lichtenbergschule Darmstadt
ca. 4 Wochen	Projekt: Datenbanken		PK3.4, PK4.4 IK1.2, IK2.1 IK2.2, IK2.4 IK3.2, IK4.1 IK4.2	z.B. LibreOffice Base; SQLsnap
ca. 3 Wochen	Klassen & Objekte	• beschreiben Basiskonzepte der objektorientierten Modellierung (Klasse, Objekt, Attribut, Methode), • entwerfen Methoden für die Manipulation von Objekten, • bilden Beziehungen zwischen Klassen ab, • unterscheiden zwischen primitiven Datentypen und Objektreferenzen.	PK1.2, PK1.3 PK2.1 IK1.4	graphische Darstellung in UML z.B. EOS2 in BlueJ

Qualifikationsphase 2. Halbjahr

Zeit-raum/ Dauer	Lernfeld/ Modul/ ggf. Thema	Lernfeldbezogene Kompetenzen	Allgemeine Kompe-tenzen	Sonstiges (Werkzeuge, An-merkungen, Bezüge, didak-tisch-methodische Hinweise)
ca. 5 Wochen	Kryptologie Asymmetri-sche Verfahren & Signatur	• erarbeiten einfache mono- und polyalphabetischen Verfahren und implementieren diese anschließend. • beschreiben das Prinzip der Transposition und der Substitution zur Verschlüsselung von Daten. • beschreiben das Prinzip der polyalphabetischen Substitution, u. a. am Beispiel des Vigenère-Verfahrens. • beurteilen die Sicherheit eines gegebenen symmetrischen Verschlüsselungsverfahrens. • entwerfen und implementieren Algorithmen unter Verwendung elementarer Zeichenkettenoperationen. • verwenden Übergabeparameter und Rückgabewerte in Operationen. • beschreiben und unterscheiden die Prinzipien der symmetrischen und asymmetrischen Verschlüsselung. • beschreiben Anwendungsbereiche für symmetrische bzw. asymmetrische Verschlüsselungsverfahren. • erläutern das Prinzip von digitalen Signaturen und Zertifikaten.	PK3.3, IK4.3	Implementierung z.B. in Python oder Java (BlueJ); Festigung der Programmier-kenntnisse, gerade im Hinblick auf Zeichenketten-Operationen und Operationen mit Parameter und Rückgabewert
ca. 5 Wochen	Automaten-modelle	• beschreiben den Aufbau und die Funktionsweise eines deterministischen endlichen Automaten (DEA). • beschreiben den Aufbau und die Funktionsweise eines endlichen Automaten mit Ausgabe (Mealy-Automat). • entwickeln und implementieren Automatenmodelle in Form von Zustandsgraphen. • analysieren die Funktion eines durch einen Zustandsgraphen vorgegebenen Automaten. • erläutern die Grenzen endlicher Automaten bei der Problemlösung.	PK1.1,PK1.5 PK2.2,PK2.3 PK3.1,PK3.2 IK3.1, IK3.2 IK4.3	AtoCC ggf. Aufbau und Funktionsweise eines Kellerautomaten bzw. einer Turingmaschine als Erweiterung des Modells des endlichen Automaten
ca. 4 Wochen	Projektarbeit	• Projekt zur Festigung und Erweiterung der Kompetenzen, z.B. Programmieren von Mikrocontrollern		z.B. Arduino Uno