

Informatik Enlightened



[1]

Verfasser: Morris Birkholz, Matthias Ehlenz, Hans-Peter Kühn, Sandra Petrut, Tobias Quix

Kurz-Info:

Informatischer Inhalt: Grundlagen der Programmierung und der technischen Informatik

Jahrgangsstufe: Mittel- und Oberstufe

Vorwissen: erste Programmierkenntnisse von Vorteil, aber nicht zwingend erforderlich

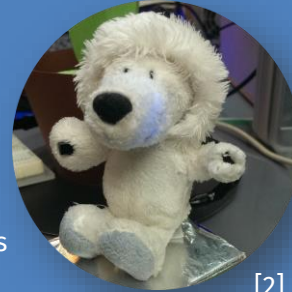
KURZINFORMATION FÜR DIE LEHRKRAFT

Titel: Informatik Enlightened

Schulstufe: Mittel- und Oberstufe

Optimale Jahrgangsstufe: Klasse 8 bis 12

Geeignete Kursart: optimal zum Ende des Wahlpflichtkurses



[2]

Themenbereich: Grundlagen der Programmierung – Kontrollstrukturen, Variablenkonzepte; Grundlagen der elektrischen Schaltungen – geschlossene Stromkreise, einfache Komponenten (Widerstände, LEDs), einfache Schaltungen

EINORDNUNG IN GESETZLICHE RAHMENBEDINGUNGEN

Kernlehrplan NRW: Kompetenzbereiche *Implementieren* und *Modellieren*, Inhaltsfelder *Informatiksysteme* und *Informatik, Mensch und Gesellschaft*

Bildungsstandards der GI: Prozessbereiche *Modellieren* und *Implementieren* und *Strukturieren* und *Vernetzen*, Inhaltsbereich *Informatik, Mensch und Gesellschaft*

Einbindung in den Unterricht: Für die Mittelstufe kann dieses Modul als Einführung in die textuelle Programmierung genutzt werden. Für einen Differenzierungskurs in der Mittelstufe, der vielleicht sogar in Kombination mit Physik unterrichtet wird, kann das Modul in einer Unterrichtsreihe über elektrische Schaltungen eingesetzt werden. Hierbei werden dann auch physikalische Hintergründe, wie z. B. geschlossene Stromkreise, betrachtet.

Die Programmiersprache des Arduino-Mikrocontrollers beinhaltet alle wesentlichen Elemente prozeduraler Programmierung, sodass dieses Modul als Einstieg in die Programmierung dienen kann. Es werden allgemeine Kontrollstrukturen und Variablenkonzepte thematisiert, die dann im weiteren Verlauf der Unterrichtsreihe vertieft werden können.

Vorkenntnisse: erste Programmierkenntnisse von Vorteil, aber nicht zwingend erforderlich

Dauer: 5 bis 6 Stunden

Inhaltsbeschreibung: Licht hat viele Facetten. Physiker würden sagen: Licht hat sowohl Wellen als auch Teilcheneigenschaften.

Auch in der Informatik wird sich mit den verschiedenen Seiten des Lichts befasst, allerdings ohne dabei so theoretisch zu werden. Denn Licht kann so viel: Licht erzeugt Energie, es überträgt Information; ob nun sichtbar oder unsichtbar begegnet es uns überall im Alltag.

In diesem Modul befassen sich die Schülerinnen und Schüler mit verschiedenen Bereichen des Themas Licht und hinterfragen, wie das alles überhaupt funktioniert. Sie erfahren, wie unsichtbares Licht beim Einparken helfen kann, wie mit Licht Geschwindigkeiten gemessen werden können, wie Informationen mittels Licht übertragen werden und wie man aus Solarzellen das Maximum herausholen kann.

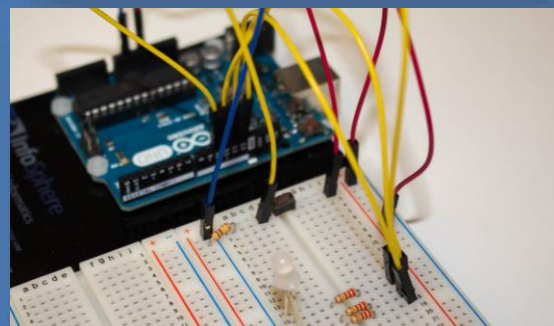
Technische Voraussetzung: Neben einem Computer/Laptop und der [Arduino-Software](#) werden im Modul InfoSphere-Kits eingesetzt, die neben dem Arduino-Mikrocontroller weitere Bauteile enthalten, die für die Schaltungen der Stationen benötigt werden. Die InfoSphere-Kits können von Lehrkräften ausgeliehen werden (schuelerlabor@informatik.rwth-aachen.de).



[3]



[4]



[5]

INHALT

Kurzinformation für die Lehrkraft.....	2
Lernziele	5
Fachliche Analyse.....	5
Einordnung in gesetzliche Rahmenbedingungen.....	8
Lehrplan Informatik.....	8
Bildungsstandards der GI	8
Benötigte unterrichtliche Voraussetzungen	8
Einbettung in den Schulunterricht	9
Didaktische/methodische Schwerpunktsetzung.....	9
Beschreibung des Modulablaufs	9
Didaktische Prinzipien nach Baumann und Hubwieser.....	9
Verlaufsplan des Moduls	11
Abbildungsverzeichnis	13
Anhang	14

LERNZIELE

Die Schülerinnen und Schüler...

- verwenden Variablen und Wertzuweisungen.
- unterscheiden verschiedene Typen von Variablen und wenden diese in ihren Programmen an.
- modifizieren und ergänzen Quelltexte von Programmen nach Vorgaben.
- interpretieren Fehlermeldungen beim Entwickeln von Software.
- verwenden bei der Implementierung algorithmische Bausteine (Schleifen, if-else-Anweisungen).
- stellen Vermutungen an über Zusammenhänge und entwickeln Lösungsmöglichkeiten im informatischen Kontext.
- beschreiben den Umgang mit Informatiksystemen aus ihrer eigenen Lebenswelt.
- implementieren auf der Grundlage von Modellen oder Modellausschnitten Computerprogramme für Arduino.
- planen und bauen einfache elektrische Schaltungen.
- beurteilen technische Geräte hinsichtlich ihres Nutzens für Mensch und Gesellschaft und ihrer Auswirkung auf die Umwelt.

FACHLICHE ANALYSE

Das Modul setzt sich aus mehreren Stationen zusammen. Station 0 wird von allen Schülerinnen und Schülern durchgeführt, wohingegen im Anschluss nur eine der Stationen 1 bis 4 bearbeitet wird. Je nach Leistungsstärke stehen den Schülerinnen und Schülern Bonus-Arbeitsblätter zur Verfügung. Auch können leistungsstarke bzw. schnelle Schülerinnen und Schüler zwei oder mehr der vier weiteren Stationen bearbeiten.

Station 0 – Einstieg

Die Station 0, die von jedem durchlaufen wird, dient als Einführung in die textuelle Programmierung und die Arbeit mit dem Arduino-Mikrocontroller. Hier werden alle benötigten Grundkenntnisse vermittelt, die auch für die weiteren Stationen relevant sind. Zu Beginn wird der Arduino-Mikrocontroller inkl. Steckbrett, seiner Anschlüsse und seiner Stromversorgung erklärt. Danach sollen mit „Informationskästen“ die Funktionen einer LED und eines Widerstands erläutert werden, die während dieser Station eine wesentliche Rolle spielen, denn die Schülerinnen und Schüler sollen diese LED zum Leuchten und Blinken bringen. Damit diese nicht durchbrennt, ist ein Widerstand wichtig. Nachdem diese grundlegenden Informationen vermittelt wurden, die auch für den Physikunterricht relevant sind, kommt die Informatik ins Spiel. Das Grundgerüst des Arduinos wird vorgestellt, indem auch Methoden thematisiert werden, die Anweisungen einmalig oder wiederholt ausführen (`setup()` und `loop()`). Da die LED an einen digitalen Pin angeschlossen werden muss, muss diese programmiert werden. Hierzu wird den Schülerinnen und Schülern vermittelt, was eine Variable ist (Vergleich zu einem Container) und wie diese deklariert bzw.

initialisiert wird. In diesem Kontext ist die Datenstruktur `int` relevant, da es bei digitalen Pins nur ganze Zahlen gibt. In diesem Zusammenhang werden auch profane Eigenschaften einer Programmierung erwähnt, wie z. B. das Semikolon (;) zum Abschließen einer Anweisung. Nach diesem kurzen Arbeitsauftrag werden Methoden benötigt, die die Schülerinnen und Schüler einfach aufrufen müssen, ohne genau zu wissen, wie diese Methoden programmiert wurden. Sie wissen jedoch, was die einzelnen Methoden benötigen, was also die Aufrufparameter sein müssen, und was sie bewirken. Die Methode, die einen Pin als digitalen Pin beschreibt, wird hierbei einmalig aufgerufen, sodass sie Platz in `setup()` findet. Dadurch kann den Schülerinnen und Schülern auch beispielhaft gezeigt werden, wann genau man Aufrufe nur einmalig benötigt. Da die LED leuchten soll (und das die ganze Zeit), muss der digitale Pin ununterbrochen eingeschaltet sein. Es ist daher sinnvoll, die Methode zum Einschalten des digitalen Pins in der `loop()` zu programmieren. Auf diese Weise erkennen die Schülerinnen und Schüler die Funktionalität einer Schleife. Ebenfalls wird in dieser Station die bedingte Anweisung benötigt und erläutert. Die LED soll eingeschaltet werden, wenn der Taster betätigt wird. Hierfür benötigt man einen analogen Anschluss für den Taster und eine Anweisung (if-Anweisung), die das ermöglicht. In einem weiteren „Informationskasten“ wird die Syntax (beispielsweise „`==`“ bei Vergleichen) einer bedingten Anweisung erklärt. Damit die LED sich auch wieder ausschaltet, brauchen die Schülerinnen und Schüler zusätzlich die `else`-Anweisung. Sie erlernen somit die komplette bedingte Anweisung. Zum Ende der Station wird der Serial Monitor vorgestellt, da auch dies ein wichtiges Werkzeug der Programmierung ist und für die weiteren Stationen benötigt wird.

Station 1 – Sonnenblume

In Station 1 soll eine Sonnenblume programmiert werden, die sich mit Hilfe eines Motors in die Richtung dreht, aus der das Licht kommt. Diese Station hat einen hohen Schwierigkeitsgrad, sodass empfohlen wird, dass sich erfahrene bzw. leistungsstarke Schülerinnen und Schüler hiermit befassen. Dies bedeutet freilich nicht, dass Schülerinnen und Schüler ohne Vorkenntnisse diese Station nicht bearbeiten dürfen. Die Materialien sind so konzipiert, dass auch Anfänger diese Station lösen können, diese jedoch länger brauchen oder vielleicht mehr Hilfe benötigen.

Das Licht wird mittels eines Helligkeitssensors erfasst, welcher an einen analogen Pin angeschlossen wird. Dies funktioniert ähnlich wie in Station 0, sodass die Schülerinnen und Schüler ihre bereits erworbenen Kenntnisse einsetzen können. Die Aufgabe ist nun, zu überprüfen, welche Werte der Helligkeitssensor bei viel und welche er bei wenig Licht misst. Dafür wird der Serial Monitor genutzt sowie eine bedingte Anweisung, die je nach gemessenem Wert sagt, ob es gerade hell oder dunkel ist. Zur Wiederholung und zur Sicherung der bedingten Anweisung wird das Grundgerüst nochmals gezeigt. Zusätzlich gibt es eine Veränderung, da die Bedingung ist, dass etwas größer bzw. kleiner als etwas anderes ist. Es wird also nicht mit zwei Gleichheitszeichen überprüft und verglichen, sondern mit den (auch aus der Mathematik bekannten) Zeichen „`>`“ und „`<`“.

Für den Servomotor wird eine zusätzliche Bibliothek benötigt, die zu Beginn des Programms eingefügt werden muss. Es wird erklärt, dass nicht jeder einprogrammierte Befehl immer

aufrufbereit ist. Dafür gibt es spezielle Bibliotheken, die man nachträglich einfügen kann und die die benötigten Befehle zur Verfügung stellen.

Station 2 – Einparkhilfe

In Station 2 soll eine Einparkhilfe, wie man sie aus herkömmlichen Autos kennt, programmiert werden. Diese Station ist vom Schwierigkeitsgrad her etwas einfacher und ist daher für unerfahrene Schülerinnen und Schüler sehr gut geeignet.

Wie auch in Station 1 besteht die erste Aufgabe darin, den IR-Distanz-Mess-Sensor auszutesten und mit Hilfe des Serial Monitors die Werte auszumessen.

Daraufhin wird der Piezo-Signalgeber eingeführt, der zur Benachrichtigung dient, wenn sich einem Gegenstand genähert wird. Um dies umzusetzen, war die Anfangsaufgabe wichtig: Die Schülerinnen und Schüler wissen hierdurch, welche Werte des Sensors ausreichend Platz oder eine Gefahr bedeuten. Außerdem wird eine bedingte Anweisung benötigt, die mittels eines „Informationskastens“ wiederholt wird. Da hier aber auch doppelte Anweisungen Einsatz finden, wird das logische Und (&&) eingeführt. Dieser logische Operator wird dann gebraucht, wenn man zwei Bedingungen hat, die gleichzeitig erfüllt sein müssen, bevor die weitere Anweisung ausgeführt wird.

Station 3 – Geschwindigkeitsmessung

In Station 3 wird eine Geschwindigkeitsmessung mittels Infrarot-LEDs und Infrarotdioden gebaut. Diese Station ist bezüglich informatischer Vorkenntnisse nicht viel schwieriger als Station 2. Allerdings ist das Bauen der Schaltung deutlich aufwendiger.

Nach dem Aufbau der Schaltung wird zu deren Programmierung übergegangen. Die Schülerinnen und Schüler benötigen auch hier Variablen für ihre Pins, und zusätzlich arbeiten sie mit einer LED, die direkt am Arduino angeschlossen ist. Für die beiden Lichtschranken benötigen die Schülerinnen und Schüler zwei bedingte Anweisungen mit je zwei Bedingungen, die mit der logischen Operation „&&“ verbunden werden. Sobald die erste Lichtschranke unterbrochen wird und die Messung noch nicht gestartet ist, soll die Messung starten. Da für eine genaue Geschwindigkeitsangabe aber auch die zweite Schranke benötigt wird, soll die Messung dann enden, wenn die zweite Lichtschranke unterbrochen wird und die Messung bereits läuft. Da dies auf den ersten Blick sehr komplex wirkt und die Schülerinnen und Schüler sogar demotivieren könnte, dient ein Programmablaufplan der Übersicht. Für die Messwerte, die beliebig groß werden können, reicht der bisher gelernte Datentyp `int` nicht mehr aus. Daher wird der Datentypen `unsigned long` eingeführt.

Station 4 – Farbthermometer

In Station 4 wird ein Farbthermometer gebaut, das mittels einer RGB-LED die Farbe verändert, sobald es wärmer bzw. kälter wird.

Zu Beginn werden die Schülerinnen und Schüler mit einer solchen RGB-LED vertraut gemacht und testen diese. Nach dem Anschließen dieser besonderen LED finden sie heraus, welche Farben diese annehmen kann und was man hierzu benötigt. Danach soll dies auch programmiert werden. Da die

RGB-LED mehr als nur zwei Beinchen hat, braucht man für die Programmierung drei Variablen, die für die einzelnen Farben (Rot, Grün und Blau) stehen. So können die Schülerinnen und Schüler jede Farbe als eigenständige LED sehen und diese dann einzeln ein- und ausschalten.

Um die RGB-LED mit der Temperatur zu steuern, wird ein Temperatursensor eingeführt, der (wie die anderen Sensoren der Stationen 1 bis 3) an einen analogen Pin angeschlossen wird. Dieser Sensor ermittelt Werte, die erst in die richtige Temperatur umgerechnet werden müssen. Auf den Arbeitsblättern wird genau beschrieben, wie das funktioniert. Außerdem wird ein weiterer Datentyp benötigt (`float`), da sowohl eine Spannung als auch eine Temperatur nicht nur ganze Zahlen annehmen kann. Um die gemessenen Werte zu sehen, soll der Serial Monitor eingeschaltet werden.

Um nun für bestimmte Temperaturen die gewünschte Farbe anzuzeigen, müssen die Schülerinnen und Schüler bedingte Anweisungen einsetzen. Da die betrachteten Temperaturbereiche sowohl eine Unter- als auch eine Obergrenze haben, wird auch in dieser Station der logische Operator `&&` erläutert.

EINORDNUNG IN GESETZLICHE RAHMENBEDINGUNGEN

LEHRPLAN INFORMATIK

Das Modul knüpft an die Kompetenzbereiche *Implementieren* und *Modellieren* an. Auch Schülerinnen und Schüler, die bisher über keine Programmierkenntnisse verfügen, können mit Hilfe des Moduls Grundlagen der Programmierung erlernen. Hierbei werden einfache Kontrollstrukturen, Methoden und Variablen erläutert, die die Lernenden nachher anhand beispielhafter Modelle (z. B. das Farbthermometer) selbst implementieren.

Im Bereich der Inhaltsfelder greift das Modul sowohl *Informatiksysteme* als auch das Themenfeld *Informatik, Mensch und Gesellschaft* auf. Die Begriffe analog und digital bzw. digitale und analoge Anschlüsse werden beschrieben, sodass die Lernenden diese anwenden können und den Unterschied erkennen. Zusätzlich tragen alltagsnahe Beispiele (Einparkhilfe, Geschwindigkeitsmessung) dazu bei, den Schülerinnen und Schülern zu vermitteln, dass ihnen Informatik nahezu überall begegnet.

BILDUNGSSTANDARDS DER GI

Dieses Modul kann mit Schülerinnen und Schülern der Mittelstufe durchgeführt werden. Auch hier knüpft es an den Inhaltsbereich *Informatik, Mensch und Gesellschaft* und den Prozessbereich *Modellieren und Implementieren* an. Zusätzlich wird der Prozessbereich *Strukturieren und Vernetzen* eingebunden, da beispielsweise auch andere Unterrichtsfächer (wie Physik) eine Rolle spielen.

BENÖTIGTE UNTERRICHTLICHE VORAUSSETZUNGEN

Für dieses Modul werden keine Vorkenntnisse benötigt. In den zur Verfügung gestellten Materialien wird alles Nötige beschrieben. Dennoch sind Basis-Programmierkenntnisse von Vorteil, da dann Informationen aus den „Informationskästen“ nur zur Wiederholung dienen. Außerdem sind auch Physik- bzw. Elektrotechnikenkenntnisse für den Bau der Schaltungen und Stromkreise von Vorteil.

EINBETTUNG IN DEN SCHULUNTERRICHT

Da dieses Modul für Schülerinnen und Schüler der Mittel- wie auch der Oberstufe geeignet ist, bieten sich viele Möglichkeiten, um es in den Schulunterricht einzubetten. Es lässt sich ebenso fächerübergreifend (z. B. in Kombination mit dem Fach Physik) wie auch im „reinen“ Informatikunterricht einsetzen. Das Modul kann als Einführung in grundlegende Regeln von Programmiersprachen dienen. Auch gesellschaftliche Aspekte finden Betrachtung. Die alltagsnahen Beispiele innerhalb der einzelnen Stationen und auch das Bauen einfacher Stromkreise verdeutlichen den Teilnehmenden, dass Informatik sehr vielfältig ist und nahezu überall im Alltag vorkommt.

Da dieses Modul keine Vorkenntnisse erfordert und Grundlagen der Programmierung vermittelt, ist es hilfreich, um die Einführung der textuellen Programmierung in der Mittel- oder auch Oberstufe zu gestalten. Die Schülerinnen und Schüler bekommen direkt einen lebensnahen Bezug zur Programmierung vermittelt, sodass auch hier die gesellschaftlichen Aspekte der Informatik nicht fehlen.

DIDAKTISCHE/METHODISCHE SCHWERPUNKTSETZUNG

BESCHREIBUNG DES MODULABLAUFS

Zum Einstieg erfolgen eine Begrüßung und eine kurze Erläuterung des Moduls und seiner einzelnen Stationen. Die Teilnehmenden arbeiten durchgehend in Zweier-Teams, was ihnen die Wichtigkeit von Kooperation in der Informatik verdeutlicht. Alle Teams starten mit der Station 0, in der alle relevanten Informationen über den Arduino-Mikrocontroller und die Programmierumgebung vermittelt werden. Grundlegende Programmierbausteine (z. B. Variablen, Kontrollstrukturen) werden ebenso erläutert wie Begriffe aus der Elektrotechnik (LED, Widerstand).

Nach der Bearbeitung der Station 0, wählt jedes Team eine weitere Station zur Bearbeitung. Die Stationen sind vom Schwierigkeitsgrad unterschiedlich, sodass auf eine Differenzierung nach der Leistungsstärke der Lernenden geachtet wird. Auch können Teams, die sehr schnell mit der Bearbeitung einer der Stationen 1 bis 4 fertig sind, sich mit Bonusmaterialien oder einer weiteren Station beschäftigen.

Zum Ende des Moduls stellen einzelne Teams ihre Lösungen vor, um auch Kompetenzen des Präsentierens zu fördern.

DIDAKTISCHE PRINZIPIEN NACH BAUMANN UND HUBWIESER

Spiralprinzip: Die Aufteilung der Arbeit während des Moduls in einzelne Stationen dient dazu, grundlegendes Wissen mit Station 0 aufzubauen und dieses mit den weiteren Stationen zu vertiefen.

Prinzip der Lebensnähe und Aktualität: Alle weiterführenden Stationen sind aktuelle Beispiele aus dem Alltag der Jugendlichen. So sind sicherlich alle schon einmal mit Einparkhilfen in Berührung bekommen.

Prinzip der Zielvorstellung: Vielen Schülerinnen und Schülern fehlt das Verständnis, wozu sie manche Inhalte lernen müssen. Die alltagsnahen Beispiele verdeutlichen den Jugendlichen, wozu sie dieses Modul und das vermittelte Wissen brauchen.

Prinzip des individuellen Lerntempos: Sobald ein Team mit Station 0 und einer weiteren Station fertig ist, kann es direkt im Anschluss Bonusmaterialien oder eine weitere Station bearbeiten. Dies fördert einerseits die leistungsstarken und die schnellen Lernenden und lässt andererseits den Leistungsschwächeren genügend Zeit, ihre Stationen zu beenden.

VERLAUFSPLAN DES MODULS

Zeit	Phase	Inhalt	Medium	Sozialform
0:00-0:05	Begrüßung	<i>Begrüßung der Schülerinnen und Schüler, Vorstellung Betreuerteam und Schülerlabor</i>		
0:05-0:20	Einstieg	<i>Vorstellung des Moduls und des Mikrocontrollers (Präsentation/Video)</i>	Surface Hub oder Beamer, Modulpräsentation, PC	LV, UG
0:20-1:30	Erarbeitung I	<i>Erarbeitung von Programmier- und elektrotechnischen Grundlagen sowie Einführung in die Arbeit mit dem Arduino-Mikrocontroller anhand von Station 0 (eine LED zum Leuchten und zum Blinken bringen mit Hilfe des Arduinos und elektrotechnischer Bauteile)</i>	AB Station 0, PC, InfoSphere-Kit	PA
1:30-1:50	Pause			
1:50-2:00	Sicherung I	<i>Vorstellung der Ergebnisse der Partnerarbeit (oder GA) zwecks Bildung gemeinsamer Wissensbasis (ca. zwei oder drei Teams)</i>	Surface Hub, Tafel oder Beamer, PC, InfoSphere-Kit	SV
2:00-4:00	Erarbeitung II	<i>Vertiefung des Grundlagenwissens: Vertiefung der Programmierkenntnisse und Erarbeitung neuer</i>	ABs zu Station 1 bis 4, PC, InfoSphere-Kit	PA/GA

<p><i>elektrotechnischer Bauteile anhand der zur Auswahl stehenden Stationen 1 bis 4</i></p> <p><i>Während dieser Bearbeitungsphase können individuelle Pausen stattfinden, sodass die Mittagspause danach auch verkürzt oder vorgezogen (bei 3:30 und danach noch weitere Bearbeitung) werden kann.</i></p>				
4:00-4:30	Mittagspause			
4:30-4:55	Sicherung II	<i>Präsentation von mindestens einem Team pro Station</i>	Surface Hub, Tafel oder Beamer, InfoSphere-Kit	SV
4:55-5:00	Verabschiedung	<i>Verabschiedung der Schülerinnen und Schüler aus dem Labor</i>		

Legende: Lehrervortrag (LV), Unterrichtsgespräch (UG), Schülervortrag (SV), Partnerarbeit (PA), Gruppenarbeit (GA), Arbeitsblatt (AB)

QUELLENVERZEICHNIS

- Gesellschaft für Informatik (GI) e. V. (2016): *Bildungsstandards Informatik für die Sekundarstufe II*. Abgerufen von: https://informatikstandards.de/fileadmin/GI/Projekte/Informatikstandards/Dokumente/Bildungsstandards_SII.pdf (eingesehen: 26.04.2022).
- Hubwieser P. (2007): *Didaktik der Informatik. Grundlagen, Konzepte, Beispiele*. 3., überarbeitete und erweiterte Auflage. Berlin und Heidelberg: Springer.
- Ministerium für Schule und Weiterbildung des Landes Nordrhein-Westfalen (Hrsg.) (2014): *Kernlehrplan für die Sekundarstufe II Gymnasium/Gesamtschule in Nordrhein-Westfalen - Informatik*. Abgerufen von: https://www.schulentwicklung.nrw.de/lehrplaene/lehrplan/75/KLP_GOST_Informatik.pdf (eingesehen: 26.04.2022).

ABBILDUNGSVERZEICHNIS

- **Abb. 1 bis 5** – Quelle: InfoSphere, CC BY-SA 4.0 Attribution-ShareAlike 4.0 International (<https://creativecommons.org/licenses/by-sa/4.0/>)

ANHANG

- Modulpräsentation
- Arbeitsblätter zu den Stationen:
 - Station 0 – Einstieg
 - Station 1 – Sonnenblume
 - Station 2 – Einparkhilfe
 - Station 3 – Geschwindigkeitsmessung
 - Station 4 – Farbthermometer
- Bonusmaterial zu den Stationen:
 - Station 2 – Einparkhilfe – Bonus
 - Station 3 – Geschwindigkeitsmessung – Bonus
 - Station 4 – Farbthermometer – Bonus 1
 - Station 4 – Farbthermometer – Bonus 2