

Seminarplan

	Inhalt	Literatur/Material für Studierende
1	Digitale Bildung in den Naturwissenschaften; 21st century skills	KMK-Strategien: Bildung in der digitalen Welt (2016); Lehren und Lernen in der digitalen Welt (2021); Voogt, J. & Roblin, N. (2010): 21st century skills Discussion Paper
2	Computational Thinking und Kompetenzmodell; Verknüpfung zur Naturwissenschaftsdidaktik	Wing, J. (2006): Computational Thinking; Senkbeil, M. et al. (2019): Das Konstrukt der Computer- und informationsbezogenen Kompetenzen und das Konstrukt der Kompetenzen im Bereich 'Computational Thinking' in ICILS 2018 .
3	Modellieren im naturwissenschaftlichen Unterricht	Nerdel, C. (2017): Grundlagen der Naturwissenschaftsdidaktik; Reiners, C. (2022): Chemie vermitteln
4	Typische digitale Tools I: Scratch	Beispielhafte Arbeitsaufträge unten beschrieben
5	Typische digitale Tools I: Scratch	
6	Typische digitale Tools II: Arduino, Python, Excel	Beispielhafte Arbeitsaufträge unten beschrieben; Materialliste und Code angegeben
7	Typische digitale Tools II: Arduino, Python, Excel	
8	Arbeit am Beispielprojekt I	Beschreibung inkl. Materialliste unten; beispielhaftes Video verlinkt
9	Arbeit am Beispielprojekt II	
10	Arbeit am Beispielprojekt III und Finalisierung Reflexionsvideos	
11	Vorstellung Beispielprodukte und Ideensammlung eigenes Projekt	
12	Arbeit am eigenen Projekt I: Vorstellung Idee und Planung	
13	Arbeit am eigenen Projekt II: Fertigstellung erster Schritte	
14	Abschlussdiskussion, Gruppeninterview	

Übungsaufgabe Computational Thinking

In der Seminarsitzung zu Computational Thinking sollten die Studierenden ein typisches mathematisches Problem mithilfe von Excel oder anderen digitalen Tools lösen. Die Studierenden erhielten bei Bedarf drei Tipps, wobei die Aufgabe wie folgt formuliert war:

Erstellt digital eine mögliche Lösung für das „Geburtstagsproblem“ (Wie viele Menschen müssen sich in einem Raum befinden, damit die Wahrscheinlichkeit, dass zwei Menschen am gleichen Tag Geburtstag haben, mindestens 50% beträgt). Dokumentiert alle nötigen Schritte und Teilüberlegungen.

Tipp 1: Betrachtet das Gegenereignis.

Tipp 2: Wahrscheinlichkeiten dürfen hier multipliziert werden

Tipp 3: Knüpft eure Überlegungen an konkrete Beispiele (Wie sieht es bei 2,3,4,... Personen aus?)

Alle Produkte wurden im Plenum unter Rückbezug auf die vorgestellten Teilkompetenzen von Computational Thinking diskutiert und Parallelen hergestellt. Eine beispielhafte Lösung einer Gruppe (vielen Dank für die Bereitstellung durch Theresa Miske und David Heindl) findet sich auf den folgenden Seiten. Die Arbeitsschritte und Überlegungen wurden durch die Gruppe während der Erstellung informell notiert und zur Verfügung gestellt.

Geburtstagsproblem

Wie viele Menschen müssen sich in einem Raum befinden, damit die Wahrscheinlichkeit, dass zwei Menschen am gleichen Tag Geburtstag haben, mindestens 50% beträgt?

Menschen in einem Raum	Tage im Jahr	Tage im Jahr		Gegenwahrscheinlichkeit	Wahrscheinlichkeit
1	365	365	1	1,0000	0
2	364	365	0,9973	0,9973	0,0027
3	363	365	0,9945	0,9918	0,0082
4	362	365	0,9918	0,9836	0,0164
5	361	365	0,9890	0,9729	0,0271
6	360	365	0,9863	0,9595	0,0405
7	359	365	0,9836	0,9438	0,0562
8	358	365	0,9808	0,9257	0,0743
9	357	365	0,9781	0,9054	0,0946
10	356	365	0,9753	0,8831	0,1169
11	355	365	0,9726	0,8589	0,1411
12	354	365	0,9699	0,8330	0,1670
13	353	365	0,9671	0,8056	0,1944
14	352	365	0,9644	0,7769	0,2231
15	351	365	0,9616	0,7471	0,2529
16	350	365	0,9589	0,7164	0,2836
17	349	365	0,9562	0,6850	0,3150
18	348	365	0,9534	0,6531	0,3469
19	347	365	0,9507	0,6209	0,3791
20	346	365	0,9479	0,5886	0,4114
21	345	365	0,9452	0,5563	0,4437
22	344	365	0,9425	0,5243	0,4757
23	343	365	0,9397	0,4927	0,5073
24	342	365	0,9370	0,4617	0,5383
25	341	365	0,9342	0,4313	0,5687
26	340	365	0,9315	0,4018	0,5982
27	339	365	0,9288	0,3731	0,6269
28	338	365	0,9260	0,3455	0,6545
29	337	365	0,9233	0,3190	0,6810
30	336	365	0,9205	0,2937	0,7063
31	335	365	0,9178	0,2695	0,7305
32	334	365	0,9151	0,2467	0,7533
33	333	365	0,9123	0,2250	0,7750
34	332	365	0,9096	0,2047	0,7953
35	331	365	0,9068	0,1856	0,8144
36	330	365	0,9041	0,1678	0,8322

Geburtstagsproblem

In unserem Seminar wurden wir mit dem Geburtstagsproblem konfrontiert. Die Aufgabenstellung war: „Wie viele Menschen müssen sich in einem Raum befinden, damit die Wahrscheinlichkeit, dass zwei Menschen am gleichen Tag Geburtstag haben, mindestens 50% beträgt?“.

Zu Beginn legten wir fest, von wie vielen Tagen im Jahr wir ausgehen. Wir haben uns auf 365 Tage geeinigt und somit das Schaltjahr außer Acht gelassen. Eine weitere Überlegung war, dass die Wahrscheinlichkeit an einem bestimmten Tag Geburtstag zu haben immer $1:365$ beträgt.

Nach einigem erfolglosen hin und her überlegen und bekamen wir den Tipp uns mit der Gegenwahrscheinlichkeit und den einzelnen Personen zu beschäftigen. Die Gegenwahrscheinlichkeit zu unserem Problem lautet in diesem Fall: „Wie viele Menschen müssen sich in einem Raum befinden, damit niemand am selben Tag Geburtstag hat?“. Da es sich um die Berechnung der Gegenwahrscheinlichkeit handelt, muss das Ergebnis auf diese Frage nachher von 1 abgezogen werden, um die eigentliche Wahrscheinlichkeit zu berechnen. Dadurch muss das Ergebnis der Berechnung des Gegenereignisses unter 0,5 liegen, um die richtige Lösung auf unsere ursprüngliche Frage zu erhalten. Da auch diese Frage von uns nicht beantwortet werden konnte, haben wir uns dazu entschlossen, das Problem in noch kleinere Teilschritte zu zerlegen. Daraus wurde die Frage: „Wie hoch ist die Wahrscheinlichkeit, dass keiner am selben Tag Geburtstag hat, wenn sich eine Person, zwei Personen, drei Personen, etc. in einem Raum aufhalten?“.

Mit dieser Frage konnten wir uns nun tatsächlich konkret auseinandersetzen. Die Überlegung war, wenn eine Person sich in einem Raum aufhält, ist die Wahrscheinlichkeit, dass keine andere Person in dem Raum am selben Tag Geburtstag hat, $365/365$. Wird dieses Ereignis berechnet, beträgt das Ergebnis 1. Befinden sich zwei Personen in einem Raum, und sollen sie an unterschiedlichen Tagen Geburtstag haben, stehen nicht mehr 365 Tage zur Verfügung, sondern nur noch $364/365$ Tage. Wird auch dieses Ereignis berechnet, ist das Ergebnis 0,9973. Bei drei Personen in einem Raum, welche an unterschiedlichen Tagen Geburtstag haben sollen, sind es nur noch $363/365$ Tage. Die Berechnung dieses Ereignisses ergibt 0,9945. Bei vier Personen in einem Raum, welche alle an unterschiedlichen Tagen Geburtstag haben sollen, stehen nur noch $362/365$ Tage zur Verfügung. Die Berechnung dieses Ereignisses ergibt 0,9918.

All diese Ereignisse haben wir von Hand berechnet und uns das Ergebnis jeweils notiert, bis uns aufgefallen ist, dass, um die Gegenwahrscheinlichkeit zu berechnen, alle Ergebnisse der Ereignisse miteinander multipliziert werden müssen. Für die ersten vier Personen machten wir diesen Schritt noch von Hand. Die Wahrscheinlichkeit, dass bei 4 Personen in einem Raum, alle an unterschiedlichen Tagen Geburtstag haben, beträgt 0,9836. Als uns klar wurde, dass diese Zahl noch sehr weit entfernt von unserem benötigten Ergebnis von unter 0,5 war und eine Menge Rechenschritte nötig waren, um dieses Ergebnis von Hand zu erreichen, machten wir uns auf die Suche nach einem einfacheren Weg.

Da ich auf Vorerfahrung in der Datenanalyse mit Excel zurückgreifen konnte, fiel mir dieses Tool als erstes ein, um immer wiederkehrende Rechenaufgaben zu lösen, ohne jeden Schritt von Hand immer wieder durchführen zu müssen. Wir beschlossen es einfach zu versuchen und gaben die Daten in ein Excel-Tabellenblatt ein, indem wir die Funktion nutzten, Reihen zu vervollständigen. Nachdem wir für die erste Zeile (eine Person im Raum) die Formeln eingegeben hatten, um die Gegenwahrscheinlichkeit und die Wahrscheinlichkeit zu berechnen, zogen wir die Formel einfach nach unten, über die anderen Zeilen hinweg. Dadurch nahm Excel uns die mühselige Rechenarbeit ab.

Das Geburtstagsproblem ist ein gutes Beispiel für die Anwendung von Computational Thinking, da ein Problem formuliert und analysiert wird. Nachdem festgestellt wird, dass sich das große Problem so nicht lösen lässt, wird es in kleinere Teilschritte zerlegt, welche nach und nach bearbeitet werden können. Nachdem eine kritische Abwägung der Vor- und Nachteile der Art der Berechnung durchgeführt wurde, werden die relevanten Daten mithilfe einer Excel Tabelle (digitales Tool) dargestellt.

Typische digitale Tools 1: Scratch

Vorab wurde den Studierenden ein kurzer Überblick über die grundlegenden Funktionen von Scratch gegeben. Als Vorbereitung auf die erste Seminarsitzung mussten die Studierenden als Übung ein typisches „Fang-die-Frucht-Spiel“ in Scratch realisieren (Beispiel: <https://scratch.mit.edu/projects/753769668>). Dieses diente zum Erarbeiten erster wichtiger Funktionsweisen von Scratch.

In den Seminarsitzungen entwickelten die Studierenden dann selbst eine Animation zu einem Diffusionsphänomen. Hierbei bestand das Ziel darin, in Abhängigkeit zur Temperatur die Verteilung eines Teefarbstoffs in Wasser durch Scratch darzustellen (Beispiel: <https://scratch.mit.edu/projects/380237241>).

Typische digitale Tools II: Arduino, Python, Excel

Vorbereitend auf die Sitzung wurden die Studierenden ebenfalls grundlegend in die Funktionsweise und den Aufbau eines Arduinos instruiert. Allerdings wurde hierbei die Arduino-Simulation von TinkerCAD (<https://www.tinkercad.com/circuits>) genutzt, um eine Hardware unabhängige Vorbereitung zu ermöglichen. Die Studierenden realisierten hierbei eine virtuelle Schaltung zur Abstandsmessung mittels eines Ultraschallsensors, welche dann zum Einstieg in die Seminarsitzung in einem realen Aufbau mittels des virtuell erstellten Codes und Schaltplans umgesetzt wurde. So konnte ein guter Einstieg in die Arbeit mit dem Arduino geschaffen werden.

Nach einer kurzen Einführung in das Konzept des mathematischen Modellierens arbeiteten die Studierenden für die weitere praktische Auseinandersetzung mit dem Arduino im Kontext des *vertical farmings*, als gesellschaftlich bedeutsames technisches Thema, mit verschiedenen Sensoren weiter. Dieses Thema diene gleichzeitig als Vorbereitung auf die Arbeit im Beispielprojekt. Die Studierenden untersuchten dabei folgende Aspekte:

- Auswirkung der Beleuchtungsdauer auf die Temperatur und Luftfeuchtigkeit
- Auswirkung des Raum-Volumens und des Abstands auf die Temperatur
- Auswirkung von eingesetzten Materialien auf die Temperatur
- Beleuchtungsintensität in Abhängigkeit zu Lichtquelle und/oder Abstand

Die Daten sammelten die Studierenden mittels Python (wurde inkl. Anleitung zur Verfügung gestellt) in einer csv-Datei und verarbeiteten sie anschließend mit Excel, um Vorhersagen treffen zu können.

Die folgende Tabelle listet das verwendete Material für eine 2er-Gruppe auf.

Arduino (oder kompatibles Board)	Verschiedene Glas- und Kunststoffgefäße
DHT11 Luftfeuchtigkeits- und Temperatursensor	Stativmaterial
LDR Fotowiderstand	Lampe mit 100W-Glühlampe
HC-SR04 Ultraschallsensor	Laptops (Eigene Geräte der Studierenden)
Breadboard und Jumperkabel	

Beispielprojekt

Anknüpfend an die Arbeit mit dem Arduino sollten die Studierenden dann in einem Beispielprojekt erste eigene Ideen entwickeln.

Als Kontext wurde das *vertical farming* fortgeführt und durch ein physisches Minibiotop ergänzt. Die Studierenden konnten dann aus folgenden Themenbereichen wählen, oder eigene Einfälle zur Erstellung eines Modells umsetzen:

1. Auswirkung der Lichtintensität auf den CO₂ Abbau
2. Skalierung der Umweltfaktoren auf größere Zusammenhänge (Entwicklung von Temperatur, Luftfeuchte, usw. in begrünten Städten o.ä.)
3. Animation zum Vorgang des CO₂ Abbaus (innerhalb der Zelle oder anderes)
4. Anschauungsmodell/ Animation zum Gasaustausch an der Blattoberfläche (Submikroskopische Ebene)

Das Ergebnis, die Arbeitsschritte und eine kurze Reflexion stellten die Studierenden dann in einem Video dar. Hier ein beispielhaftes Video von Josefine Elzner und Marie-Lina Weber:

<https://youtu.be/yVggvFJzTgQ>

Ergänzend zur Materialliste von oben wurde für das Beispielprojekt folgendes Material zusätzlich eingesetzt:

CCS811 CO ₂ -Sensor	Becherglas
Kunststoffkiste mit Deckel (40cm x 27cm x 15,5cm)	Citronensäure, Natron, Wasser
Farn	

Zur Verfügung gestellter Code

Da beinahe alle Studierenden der ersten Seminardurchführung keine Vorkenntnisse mit Arduino aufweisen konnten, wurden die Codes der einzelnen Sensoren bei Bedarf zur Verfügung gestellt. Die Kombination mehrerer Sensoren oder alternativer Anschlüsse mussten die Studierenden selbst vornehmen.

Code für DHT11:

```
#include "DHT.h" //Einbinden der DHT-Bibliothek. Muss vorher geladen werden. Ladet das Paket "DHT-Sensors" von
"Adafruit"
#define DHTPINL2 7 //Anschluss-PIN
#define DHTTYPE DHT11 //DHT-Typ. Es gibt noch einen genaueren (aber teureren) DHT22
DHT _dht_L2(DHTPINL2, DHTTYPE);
float temp=0; //Defintion Variable Temperatur
float hum=0; //Definition Variable Luftfeuchtigkeit

void setup()
{
  Serial.begin(9600); //Starten der seriellen Verbindung, um Werte aus dem seriellen Monitor anzeigen zu lassen
  _dht_L2.begin(); //Starten der Kommunikation mit dem Sensor
  //Ausgeben einer Startzeile
  Serial.print("Temperatur");
  Serial.print(";");
  Serial.println("Luftfeuchtigkeit");
}

void loop()
{
  temp=_dht_L2.readTemperature(); //Lesen der Temperatur und speichern unter Variable temp
  hum=_dht_L2.readHumidity(); //Lesen der Luftfeuchtigkeit und speichern unter Variable hum
  //Ausgabe der Daten
  Serial.print(temp);
  Serial.print(";");
  Serial.println(hum);
  delay(2000); //DHT-Sensoren sind recht langsam. Deshalb längere Pause Pflicht.
}
```

Code für HC-SR04:

```
// Variable zur Speicherung der gemessenen Entfernung
float distance = 0;

// Funktion zum Auslesen der Ultraschall-Entfernung
// Übergabe der Trigger- und Echo-Pins
long readUltrasonicDistance(int triggerPin, int echoPin)
{
    // Trigger-Pin als Ausgang festlegen und auf LOW setzen, um den Trigger zu löschen
    pinMode(triggerPin, OUTPUT);
    digitalWrite(triggerPin, LOW);
    delayMicroseconds(2);

    // Den Trigger-Pin für 10 Mikrosekunden auf HIGH setzen
    digitalWrite(triggerPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(triggerPin, LOW);

    pinMode(echoPin, INPUT); // Echo-Pin als Eingang festlegen

    return pulseIn(echoPin, HIGH); // Den Echo-Pin auslesen und die Laufzeit des Schallimpulses in
    Mikrosekunden zurückgeben
}

void setup() {
    Serial.begin(9600); // Serielle Kommunikation mit einer Baudrate von 9600 starten
}

void loop() {
    distance = 0.01723 * readUltrasonicDistance(7, 6); // Entfernungsmessung durchführen und das Ergebnis in der
    Variable "distance" speichern

    Serial.println(distance); // Die gemessene Entfernung über die serielle Schnittstelle ausgeben

    delay(2000); // Eine Pause von 2 Sekunden einlegen
}
```

Code zum Auslesen der Werte des LDR Fotowiderstands:

```
int eingang= A0; //Das Wort „eingang“ steht jetzt für den Wert „A0“ (Bezeichnung vom Analogport 0)
long sensorWert = 0; //Variable für den Sensorwert mit 0 als Startwert
long sensorWertmin = 700; //Mindestwert zum Kallibrieren des Lichtsensors
long sensorWertmax = 900; //Maximalwert zum Kallibrieren des Lichtsensors
float helligkeit=0; //Helligkeit als Kommazahl mit Startwert 0

void setup() {
    Serial.begin(9600); //Die Kommunikation mit dem seriellen Port wird gestartet. Das benötigt man, um sich den
    tatsächlich ausgelesenen Wert später im seriellen Monitor anzeigen zu lassen
}

void loop() {
    sensorWert =analogRead(eingang); //Die Spannung an dem Fotowiderstand auslesen und unter der Variable
    „sensorWert“ abspeichern.
    if (sensorWert < sensorWertmin) {sensorWertmin = sensorWert;} else {sensorWert=sensorWert;}
    //Kallibrierung des Lichtsensors
    if (sensorWert > sensorWertmax) {sensorWertmax = sensorWert;} else {sensorWert=sensorWert;}
    //Kallibrierung des Lichtsensors
    helligkeit= map(sensorWert, sensorWertmin, sensorWertmax,0,100); //Zuweisung des Sensorwerts an einen
    Wert zwischen 0 und 100 für die Helligkeit
    //Ausgabe der Werte auf dem seriellen Monitor
    Serial.print(sensorWert);
    Serial.print(";");
    Serial.print(helligkeit);
    Serial.print(";");
    Serial.print(sensorWertmin);
    Serial.print(";");
    Serial.println(sensorWertmax);

    delay (250); //Eine kurze Pause, um keine „Datenflut“ zu erzeugen
}
```

Code für CCS811:

#include"Adafruit_CCS811.h" //Einbinden der CCS811-Bibliothek. Muss vorher geladen werden. Ladet das Paket "CCS811" von "Adafruit". Mögliche automatische Zusatzinstallationen akzeptieren.

Adafruit_CCS811 ccs;

void setup() {

 Serial.begin(9600);

 Serial.println("CCS811 test");

 Serial.println("CO2-Gehalt in ppm");

 if(!ccs.begin()){

 Serial.println("Failed to start sensor! Please check your wiring.");

 while(1);

 }

// Warten, bis der Sensor bereit ist while(!ccs.available());

}

void loop() {

 if(ccs.available()){

 if(!ccs.readData()){

 Serial.println(ccs.getCO2());

 }

 } else{

 Serial.println("ERROR!");

 while(1);

 }

 }

 delay(2000);

}

Python-Code zum Auslesen der seriellen Schnittstelle:

```
import serial #Muss vorher installiert werden mit 'pip install pyserial'
import os

# Serielle Verbindung konfigurieren (COM-Port, Baudrate usw.)
ser = serial.Serial('COM6', 9600) # COMx durch den richtigen COM-Port ersetzen (Mit Arduino auslesen)

# Pfad zur Desktop-Verzeichnis abrufen
desktop_path = os.path.expanduser("~/Desktop")

# Datei zum Schreiben öffnen
file_path = os.path.join(desktop_path, "output.txt") #Dateiname kann von output.txt geändert werden
with open(file_path, 'w') as file:
    # Schleife zum Empfangen und Schreiben der Daten
    while True:
        if ser.in_waiting > 0:
            # Daten vom Seriellen Monitor lesen
            data = ser.readline().decode().strip()
            # Daten in Textdatei schreiben
            file.write(data + '\n')
            # Daten auf der Konsole ausgeben (optional)
            print(data)

# Serielle Verbindung schließen
ser.close()
```