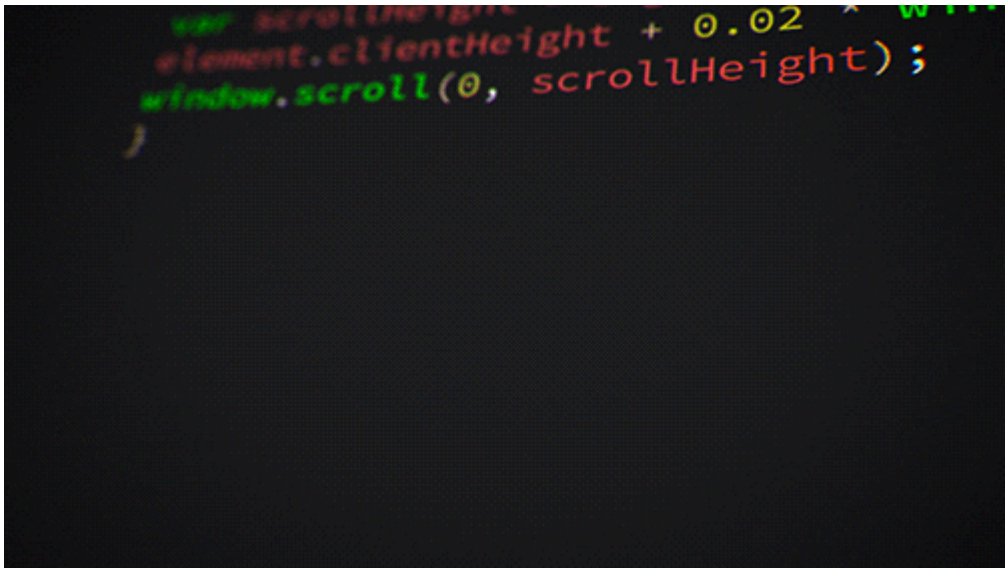
Organisatorische Einordnung / Motivation

Das vorliegende Material bedient die bisherigen Veranstaltungen

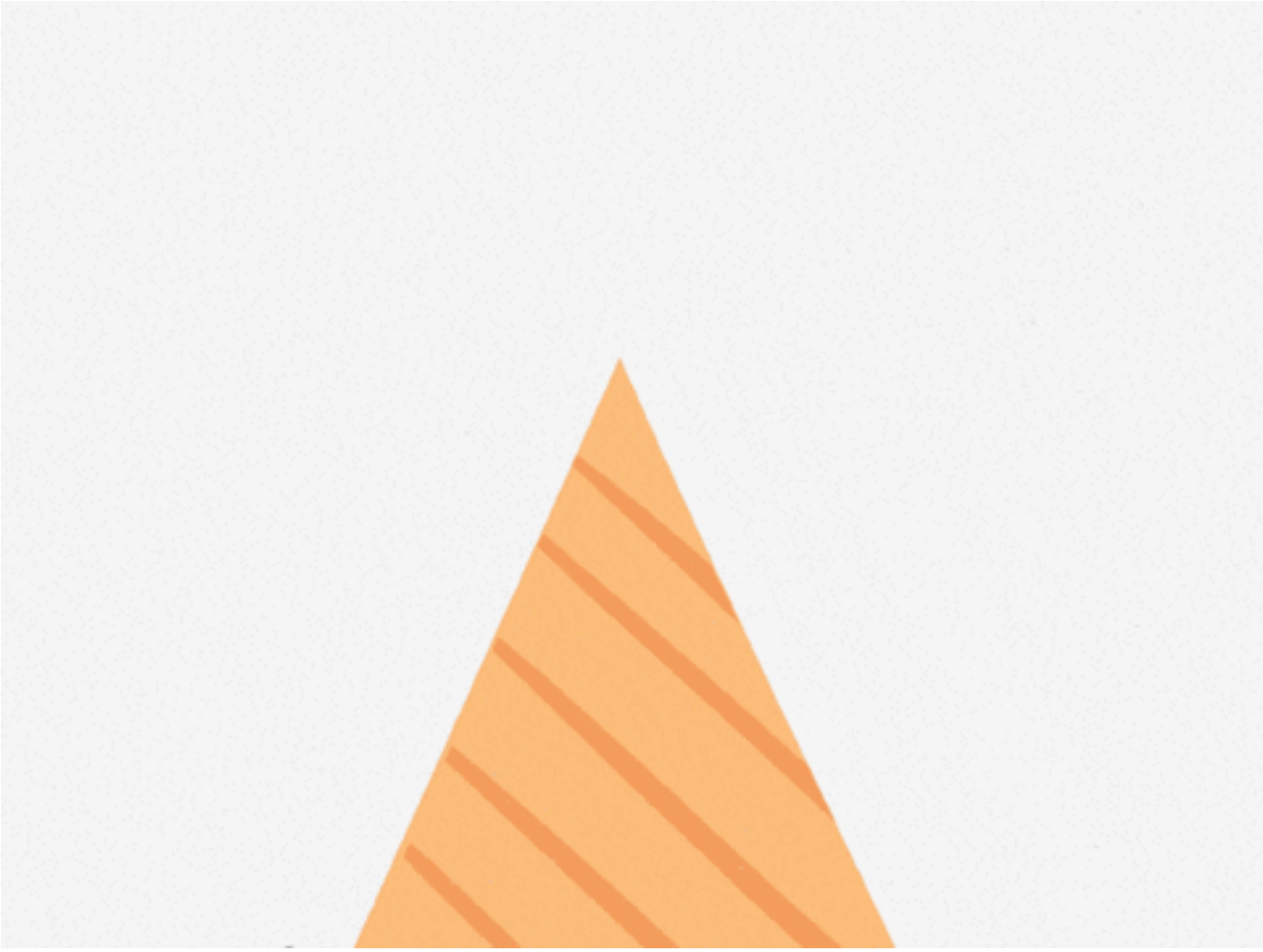
- *Prozedurale Programmierung* und
- *Einführung in die Informatik*

in einer übergreifenden Vorlesung. Diese zielt auf die Vermittlung grundlegender Konzepte und anwendungsbezogener Fähigkeiten beim Entwurf und der Umsetzung von Softwarelösungen im Kontext der wissenschaftlichen Arbeit.

Die Veranstaltung richtet sich an Nicht-Informatiker aus verschiedenen ingenieurwissenschaftlichen Disziplinen mit keinen oder geringen Programmierkenntnissen.



Ausgangspunkt



Ein Blick ins Modulhandbuch

Ziele:

- Studierende sollen verstehen, was Algorithmen sind und welche Eigenschaften sie haben,
- in der Lage sein, praktische Probleme mit wohl strukturierten Algorithmen zu beschreiben,
- die Syntax und Semantik einer prozeduralen Programmiersprache beherrschen, um Algorithmen von einem Computer erfolgreich ausführen zu lassen,
- Datenstrukturen und algorithmische Konzepte kennen und über Wissen ausgewählter Standardalgorithmen verfügen.

Inhalte: Grundlegende Prinzipien und Eigenschaften von Algorithmen und deren prozedurale Programmierung:

- Datentypen und Variablen
- Zeiger und Felder
- Anweisungen
- Ausdrücke
- Operatoren
- Kontrollstrukturen
- Blöcke und Funktionen
- Strukturen
- Typnamen und Namensräume
- ...

Konkrete Anwendungsszenarien

Allgemein	**Beispiel**
1. Erstellen einer Hypothese	*Der freie Fall ist eine beschleunigte Bewegung.*
2. Entwurf eines Experimentes für die systematische Untersuchung der Fragestellung	*Wir lassen eine Kugel aus verschiedenen Höhen fallen und messen die Fallzeit bis zum Aufschlag. Wir ignorieren Einflüsse der Luftreibung.*
3. Durchführung des Experimentes	*?*
4. Bewertung der erlangten Daten	*?*

Merke: Wissenschaftliches Arbeiten ist im Bereich der Natur- und Ingenieurwissenschaften ohne den Rechner (fast) nicht denkbar.

Beispiel

1. Hypothese: Am Wochenende scheint die Sonne häufiger als unter der Woche.

2. Konzeption eines Experimentes: ...

Frage: Wie würden Sie vorgehen?

3. Durchführung des Experimentes

4. Filterung und Interpretation der erlangten Daten

Das Diagramm zeigt die Darstellung der Lichtintensität über den Stunden eines Tages für eine Woche. Blau sind die Wochentage markiert, rot der Samstag und der Sonntag.

Frage: Welche Kritik sehen Sie an dieser Methodik?

Unsere Motivation

- **Anwendungssicht**

  Wir möchten Sie in die Lage versetzen einfache Messaufgaben mit einem Mikrocontroller zu realisieren und die Daten auszuwerten.

- **Algorithmische Perspektive**

  Wir möchten Sie dazu ertüchtigen den Algorithmusbegriff der Informatik zu durchdringen und anwenden zu können.

- **Konzeptionelle Perspektive**

  Sie erlernen grundlegende Elemente der prozeduralen und der objektorientierten Programmierung.

- **Umsetzungssicht**

  Wir vermitteln Grundkenntnisse in den Programmiersprachen C++ und Python.

Zwischenfrage: Welche Argumente vermuten Sie hinter der Entscheidung zwei Programmiersprachen in die Vorlesung aufzunehmen?

	Phase 1	**Phase 2**
Programmiersprache	C++	Python
Framework / Packages	Arduino	pandas/numpy/matplotlib
Ziel	Datenerhebung	Datenauswertung
Plattform	(PC) Mikrocontroller	PC

Keine Angst vor Code!

Jetzt mal konkret

Das Beispiel zeigt ein "Hello World" Beispiel für die C++ Implementierung auf dem Mikrocontroller. Um das einfachste Beispiel darzustellen, lesen wir keinen Sensor ein, sondern lassen eine kleine LED blinken.

Hello World - Beispiele Phase 1

Mit einiger Erfahrung C++ lassen sich einfache Mess- und Regelungstechnikaufgaben sehr einfach automatisieren. Ein Startpunkt dafür ist das Arduino Projekt, dass sowohl eine entsprechende Hardware, wie auch eine Programmierumgebung bereitstellt.

helloWorldinArduino.cpp

```
1  void setup() {
2    pinMode(LED_BUILTIN, OUTPUT);
3  }
4
5  void loop() {
6    digitalWrite(LED_BUILTIN, HIGH);
7    delay(1000);
8    digitalWrite(LED_BUILTIN, LOW);
9    delay(1000);
10 }
```

```
Sketch uses 924 bytes (2%) of program storage space. Maximum is 32256
bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes
for local variables. Maximum is 2048 bytes.
```

Die Programmiersprache Python offeriert eine Vielzahl von Paketen für unterschiedlichste Zwecke. Wir werden uns auf die Visualisierung und Datenauswertung konzentrieren.

Hello World - Beispiele Phase 2

```python
import matplotlib.pyplot as plt
import numpy as np

N_points = 100000
n_bins = 20

rng = np.random.default_rng(19680801)
dist1 = rng.standard_normal(N_points)

fig1, ax1 = plt.subplots()
ax1.hist(dist1, bins=n_bins)
ax1.set_title('Distribution')
ax1.set_ylabel('Count')
ax1.set_xlabel('Value')

fig1
```

"Python ist ja nett, aber C++ ist ..."

Foreneinträge aus Veranstaltungen anderer Hochschulen:

> *"Viele haben bei uns wegen dem Info-Grundlagenmodul gewechselt. Allerdings hängt das auch von dir und deinem Talent ab. Das Tempo ist rasant. Jede Art von Vorerfahrung hilft dir eigentlich sehr. Also wenn du noch Zeit hast vorm Studienbeginn, schnapp dir ein gutes Buch zur gelehrten Sprache, und fange schonmal bissl an kleine Sachen zu programmieren."*
>
> *"Ich habe es gerade irgendwie selbst gelöst, aber keine Ahnung warum es funktioniert hat."*

Frage: Was sind die besonderen Herausforderungen bei der Programmierarbeit?

- Abstrakte Denkweise
- Penible Beachtung der Syntax
- Ungewohnte Arbeitsmittel

BuggyCode.cpp

```
 1 void setup() {
 2   pinMode(LED_BUILTIN, OUTPUT)
 3 }
 4
 5 void loop() {
 6   digitalWrite(LED_BUILTIN, HIGH);
 7   delay(1000);
 8   digitalwrite(LED_BUILTIN, LOW);
 9   delay(1000);
10
```

```
/sketch/sketch.ino: In function 'void setup()':
/sketch/sketch.ino:3:1: error: expected ';' before '}' token
 }
 ^
/sketch/sketch.ino: In function 'void loop()':
/sketch/sketch.ino:8:3: error: 'digitalwrite' was not declared in this
scope
   digitalwrite(LED_BUILTIN, LOW);
   ^~~~~~~~~~~~
/sketch/sketch.ino:8:3: note: suggested alternative: 'digitalWrite'
   digitalwrite(LED_BUILTIN, LOW);
   ^~~~~~~~~~~~
   digitalWrite
/sketch/sketch.ino:9:14: error: expected '}' at end of input
   delay(1000);
              ^

Error during build: exit status 1
```

Wo stehen Sie?

Wer von Ihnen hat bereits Programmiererfahrung? Was würden Sie auf die Frage antworten, wieviele Codezeilen Sie bereits geschrieben haben?

- ◯ Keine einzige Zeile
- ◯ 20 Zeilen in der Schule
- ◯ n Zeilen in eigenen Programmierprojekten
- ◯ sehr viele Zeilen in Open Source Projekten

Organisatorisches

Wer sind *wir*?

Name	Email	Rolle
Prof. Dr. Sebastian Zug	sebastian.zug@informatik.tu-freiberg.de	Vorlesungen
Dr. Galina Rudolf	galina.rudolf@informatik.tu-freiberg.de	Vorlesungen und Koordination der Übungen
Nico Sonack	nico.sonack@student.tu-freiberg.de	Tutor
Florian Richter	Flo.Richter@informatik.tu-freiberg.de	
Robert Lösch	Robert.Loesch@informatik.tu-freiberg.de	
Maximilian Petry	Maximilian.Petry@student.tu-freiberg.de	Tutor

Strukurierung der Veranstaltung

Nr.	Datum		Inhalt
0	18.10.2022		*Dies Academicus*
1	25.10.2022	C++	Motivation, Organisation
2	01.11.2022	C++	C++ Programmstrukturen / Entwicklungsprozess
3	07.11.2022	C++	Datentypen / Ein- und Ausgabe
4	15.11.2022	C++	Kontrolstrukturen
5	22.11.2022	C++	Zeiger und Arrays
6	29.11.2022	C++	Funktionen
7	06.12.2022	C++	Objekte
8	13.12.2022	C++	Vererbung
9	20.12.2022	C++	Anwendung auf dem Mikrocontroller
10	10.01.2023	Python	Python Grundlagen
11	17.01.2023	Python	Objekte
12	24.01.2023	Python	Visualisierung
13	31.01.2023	Python	Datenanalyse
14	07.02.2023	Python	Übergreifende Anwendungen

Achtung: C++ wird zunächst ohne die spezifische Verwendung des Mikrocontrollers verwendet. Vielmehr erfolgt die Programmierung auf dem Desktoprechner. Wenn wir die Grundlagen erarbeitet haben, wechseln wir die Hardware einfach aus.

Übungen

Die Übungen vertiefen das erlernte anhand praktischer Programmieraufgaben:

... haben wir hier noch mal ein Beispiel?

> **Frage:** Ist jemand noch nicht für die Übungen eingeschrieben?

> **Empfehlung:** Für die Arbeit am Programmcode empfehlen wir den freien Editor Visual Studio Code. Dieser kann sowohl für die Arbeit mit den C++ als auch Python Programmbeispielen genutzt werden.

Vorlesungsmaterialien

Die Vorlesungsunterlagen selbst sind unter

https://github.com/SebastianZug/VL_ProzeduralePRogrammierung

verfügbar. Diese können entweder in der Markdown-Syntax oder als interaktives Dokument betrachtet werden.

> Achtung! Es handelt sich um "lebende" Materialien.
>
> - der Inhalt wird sich ggf. anhand Ihrer Verbesserungsvorschläge verändern
> - die Dokumente enthalten eine Vielzahl von ausführbarem Code.

Erwartungen

Der Zeitaufwand beträgt 180h und setzt sich zusammen aus 60h Präsenzzeit und 120h Selbststudium. Letzteres umfasst die Vor- und Nachbereitung der Lehrveranstaltungen, die eigenständige Lösung von Übungsaufgaben sowie die Prüfungsvorbereitung.

Sie müssen, insbesondere wenn Sie noch keine Programmiererfahrung haben, ggf. deutlich mehr Zeit in den Kurs investieren. Dies macht gemeinsam mehr Spaß als allein!

Wie können Sie zum Gelingen der Veranstaltung beitragen?

- Stellen Sie Fragen, seinen Sie kommunikativ!

> Hinweis auf OPAL Forum!

- Organisieren Sie sich in Arbeitsgruppen!

Hinweis auf Repl.it

- Machen Sie Verbesserungsvorschläge für die Vorlesungsfolien!
- Sprechen Sie uns gern wegen "Bastelbedarf" für ein eigenes Projekt an!

Beispiel der Woche

```
#include "Sensor.h"
#include "RGB_LED.h"

RGB_LED rgbLed;
DevI2C *i2c;
LSM6DSLSensor *sensor;
int xAxesData[3];
int prev_pos;

void setup() {
    Serial.begin(9600);//Baudrate
    i2c = new DevI2C(D14,D15);
    sensor = new LSM6DSLSensor(*i2c,D4,D5);
    sensor -> init(NULL);
    sensor -> enableAccelerator();
}

void loop() {
    sensor ->getXAxes(xAxesData);
    Serial.printf("%d %d %d\n",xAxesData[0],xAxesData[1],xAxesData[2]);
    if (xAxesData[2]<1000){
      rgbLed.setColor(255,0,0);
    }else{
      rgbLed.turnOff();
    }
    delay(50);
}
```