

Hardware I & Programmieren I

Erwartete Note: _____

Punkte: _____ / 30 Note: _____

Teil 1: Auf Papier

Aufgabe 1: Cores & Threads / 3 P.

Betrachte ein Computer mit 2 Kernen an jeweils 4 Threads und einer Taktfrequenz von 2 GHz.

(a) (2 P.) Berechne die Anzahl Operationen pro Sekunde.

Lösung:

$$2 \cdot 4 \cdot 2 \cdot 10^9 = 16\,000\,000\,000$$

(b) (1 P.) In welcher Komponente werden diese Berechnungen ausgeführt?

Lösung: CPU

Aufgabe 2: Schnittstellen / 9 P.

(a) (3 P.) Benenne drei **Benutzerschnittstellen** eines Computers?

Lösung:

- Display
- Touchpad
- Mikrofon
- Audio-Ausgabe
- Vibration
- ...

(b) (4 P.) Betrachte das Bild:



Welche Schnittstellen sind abgebildet?

Lösung:

1. LAN (RJ-45) Netzwerkbuchse
2. Audio-Jack (Audio Out / Mikrofon)
3. USB-C
4. USB-A

- (c) (2 P.) Betrachte die mit 4 nummerierte Schnittstelle. Benenne die Hauptverwendungszwecke der Schnittstelle. Was sind die Unterschiede zur mit 3 nummerierten Schnittstelle?

Lösung:

1. Geräte verbinden (Maus, Tastatur, Speicherstick, Telefon)
2. Display / LAN
3. Stromversorgung für kleine und grosse Geräte (Telefon, PC...)
4. Kann kein Display betreiben im Gegensatz zu USB-C.

Aufgabe 3: Speicher / 7 P.

- (a) (2 P.) Du kaufst eine neue Festplatte, um deine Fotos und Filme darauf zu speichern. Du hast 25000 Fotos und 30 FullHD-Filme. Wie gross sollte die Festplatte sein? Triff eine sinnvolle Annahme für die Grösse eines Fotos und eines Films

Tip: Verwende Zahlen, die sinnvoll und *einfach zum rechnen* sind.



Lösung: Annahme: 1 Foto: 4 MB, 1 Film: 5 GB, also 5000 MB

$$25000 \cdot 4 \text{ MB} + 30 \cdot 5000 \text{ MB} = 250\,000 \text{ MB} = 250 \text{ GB}$$

- (b) (3 P.) Das abgebildete iPhone wird mit zwei Speicher-Arten beworben. Welcher Speicher ist mit *Speicherkapazität* gemeint? Erkläre kurz den Unterschied zwischen den beiden.



849.-

Apple iPhone 12 (128 GB, Black, 6.10", SIM + eSIM, 12 Mpx, 5G)

★★★★★

Datenspeicher Eigenschaften

Arbeitsspeicher	4 GB
Speicherkapazität	128 GB

Lösung:

- Arbeitsspeicher (RAM, Hauptspeicher): sehr schnell, Daten gehen verloren, wenn der Strom wegfällt.
- Permanentenspeicher: langsamer, günstiger, Daten bleiben gespeichert bei Stromausfall.

- (c) (2 P.) Ein Taschenrechner-Programm soll die Benutzereingabe abspeichern. Der Taschenrechner hat Tasten für die Ziffern sowie die Operatoren der Grundrechenarten (+ − × ÷). Wieviele verschiedene Tasten gibt es? Wieviele Bits werden benötigt, um einen einzelnen Tastendruck zu speichern? Zei Der Rechnungsweg muss klar ersichtlich sein.

Lösung: Es gibt 10 Ziffern und 4 Operatoren, also 14 verschiedene Eingaben. Dafür werden 4 bit benötigt:

$$2^3 = 8$$

$$2^4 = 16$$

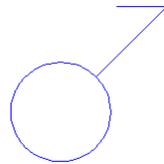
Teil 2: Am Computer

Speichere regelmässig!!!

- Speichere das Programm für jede Aufgabe als `vorname_a4.py` bzw. `vorname_a5.py`.
- Füge die gespeicherte Datei an die Aufgabe in Teams hinzu.
- Am Schluss die Teams-Aufgabe *abschliessen*.
- Falls es Probleme gibt, kannst du die Programme auch per Email senden.

Aufgabe 4: Turtle 101 / 5 P.

Schreibe ein Programm, welches die folgende Figur möglichst genau reproduziert. Die Verwendung von `repeat` ist hier *nicht* nötig.



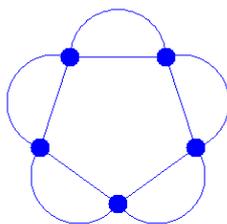
Lösung: Bewertung: Funktioniert \pm : 5P; Abzug für Syntaxfehler, ungenaue Winkel, fehlende Elemente. Bsp: Nur Kreis: 2P (Pfeil fehlt, Spitze fehlt, je 1P Abzug)

```
1 from turtle import *
2
3 g = Turtle()
4 g.ht()
5
6 g.left(45)
7 g.circle(50,360)
8 g.right(90)
9 g.forward(100)
10 g.left(135)
11 g.forward(50)
12 g.back(50)
13 g.left(90)
14 g.forward(50)
```

Aufgabe 5: Pentaflower / 6 P.

Schreibe ein Programm, welches die folgende Figur möglichst genau reproduziert. Für die volle Punktzahl sind folgende Vorgaben notwendig:

- Verwende wo möglich den `repeat`-Befehl.
- Keine Linie wird mehr als einmal abgefahren (aber Punkte dürfen mehrfach berührt werden).
- Keine Sprünge (die Figur wird in einem Stück ohne Absetzen gezeichnet).



Lösung: Bewertung: Funktioniert \pm : 4P; Abzug für Syntaxfehler, ungenaue Winkel, fehlende Elemente. Zusatzpunkte: 1P für *repeat*; 1P total für zusätzliche Bedingungen.

```
1 from turtle import *
2
3 t = Turtle()
4 t.ht()
5
6 n = 5
7
8 t.left(90)
9
10 repeat n:
11     t.dot(20)
12     t.forward(100)
13     t.left(360/n)
14
15 t.right(90)
16
17 repeat n:
18     t.circle(50,180)
19     t.right(180-360/n)
```