1 Programmierkonzepte (unplugged)

1.1 Steuerung und Regelung

⚒ **Aufgabe A1** Im Schulzimmer wird eine «Ladestation» definiert. Diese muss ein «Roboter» aufsuchen, um sich wieder aufzuladen.

Der Roboter wird von einem Schüler gespielt. Dessen Aufgabe ist es, die erhaltenen Befehle möglichst exakt so umzusetzen (und nicht zwingend so, wie die Befehle wohl gemeint waren). Unsinnige oder gefährliche Befehle dürfen auch einfach mit «Fehler: Unbekanter Befehl» quittiert werden.

a) Der «Programmierer» erteilt dem Roboter mündlich möglichst wenige kurze, einfache und eindeutige Befehle, um den Roboter zur Ladestation zu führen.

b) Arbeiten Sie in Zweiergruppen, wobei eine Person den Roboter spielt und die andere für die Programmierung zuständig ist. Legen Sie eine Startposition des Roboters fest und geben Sie dem Roboter einen Zettel mit schriftlichen Anweisungen, die dieser dann Schritt für Schritt abarbeitet. Ziel ist immer noch, dass der Roboter den Weg zur Ladestation findet.
Tauschen Sie dann die Rollen und verbessern/erweitern Sie das Programm (d.h. die schriftlichen Anweisungen).

c) Damit nicht jeder Roboter einen eigenen Befehlssatz hat, möchten wir einen «standardisierten Befehlsatz» ausmachen. Der soll einerseits «universell» genug sein, aber auch möglichst klein.

Befehl	Wirkung

d) Versuchen Sie, eine Roboterprogrammierung aufzuschreiben, mit der ein Roboter den Weg finden kann, egal wo er startet und in welche Richtung er anfangs schaut.
Eventuell müssen wir dafür den Befehlssatz noch erweitern. Schlagen Sie Ihre Erweiterungen der Klasse vor.

e) Wer hat das «kürzeste» universelle Programm (i.e. am wenigsten Befehle)?

f) Wer hat das «effizienteste» Programm (i.e. jenes, mit der der Roboter im Durchschnitt am schnellsten am Ziel ist)?

Merke 1.1.1 Steuerung vs. Regelung

In Aufgabe A1 wurde der Roboter erst in Echtzeit ferngesteuert, dann für einen fixen Weg vorausprogrammiert. Der Roboter wurde **gesteuert**. Das Programm kann nicht auf Unvorhergesehenes oder neue Startpositionen reagieren.
Am Schluss wurde ein universales Programm entworfen, das Unvorhergesehenes reagieren kann. Man spricht dann von **Regelung**.

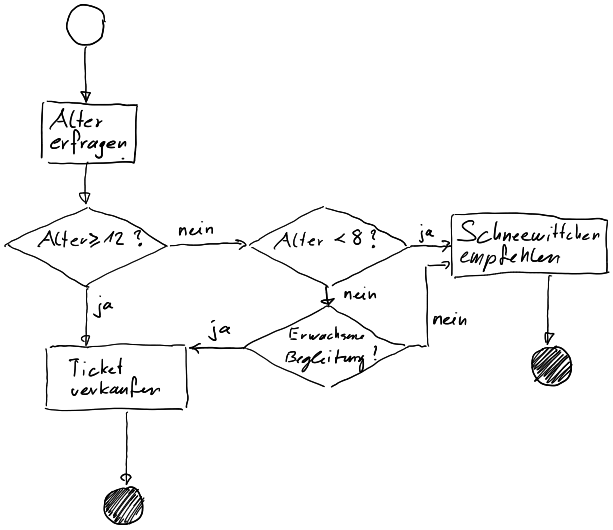
Aufgabe A2 Sind folgende Dinge eher gesteuert oder geregelt? Argumentieren Sie!

a) Liftposition bezüglich Stockwerk b) Lifttüren c) Heizung
d) Herdplatte e) Kühlschrank f) Wohnzimmerbeleuchtung

1.2 Flussdiagramme

Flussdiagramme sind zwar sehr einfach zu verstehen, für komplexere Programme oder Abläufe aber eher ungeeignet und darum kaum gebräuchlich, was mit folgenden Aufgaben begreifbar gemacht werden soll.

Aufgabe A3
Sie haben sich für einen Ferienjob an der Kinokasse beworben und diesen auch erhalten. Am ersten Tag verkaufen Sie nur Billete für den Film «Flippper im Flussdiagramm». Dazu erhalten Sie die Skizze rechts als Anweisung. Formulieren Sie diese Einlassregel in zwei einfachen Sätzen.



Merke 1.2.1 Elemente eines Flussdiagramms

In obigem Flussdiagramm haben wir 4 Elemente:

Start Symbolisiert durch einen leeren Kreis.

Anweisung Rechteck mit genau einem ausgehenden Pfeil.

Verzweigung Raute mit Bedingung und zwei Ausgängen «ja» und «nein»

Ende Symbolisiert durch einen vollen Kreis.

Aufgabe A4 Sie haben den ersten Tag Ihres Ferienjobs an der Kinokasse gut gemeistert und verkaufen nun Tickets zu drei Filmen:

- «Schneewittchen» ohne Altersbegrenzung.
- «Flipper im Flussdiagramm» (wie in Aufgabe A3).
- «Der Spaghetti-Code» ab 16 Jahren, (in Begleitung ab 13 Jahren).

Zeichnen Sie den Ablauf als Flussdiagramm.

Aufgabe A5 Im Kino gibt es einen Getränke-Automat, der nur Mineralwasser der Marke «Trivial» zum Preis von CHF 3.50 verkauft.

a) Beschreiben Sie das Verhalten vom Getränkeautomat mit einem Flussdiagramm. Das Diagramm soll keinen Endknoten haben, sondern der Automat wartet einfach auf den nächsten Verkauf.
b) Testen Sie Ihr Flussdiagramm in Zweiergruppen und ergänzen/korrigieren Sie dieses falls nötig.

c) Der Automat wird nun auf 3 Getränke ausgebaut, es kommen «Trivial Orange» für CHF 4.- und «Trivial Cola» für CHF 4.50 dazu. Zeichnen Sie das Flussdiagram und testen Sie es wieder zu zweit.

1.3 Variablen

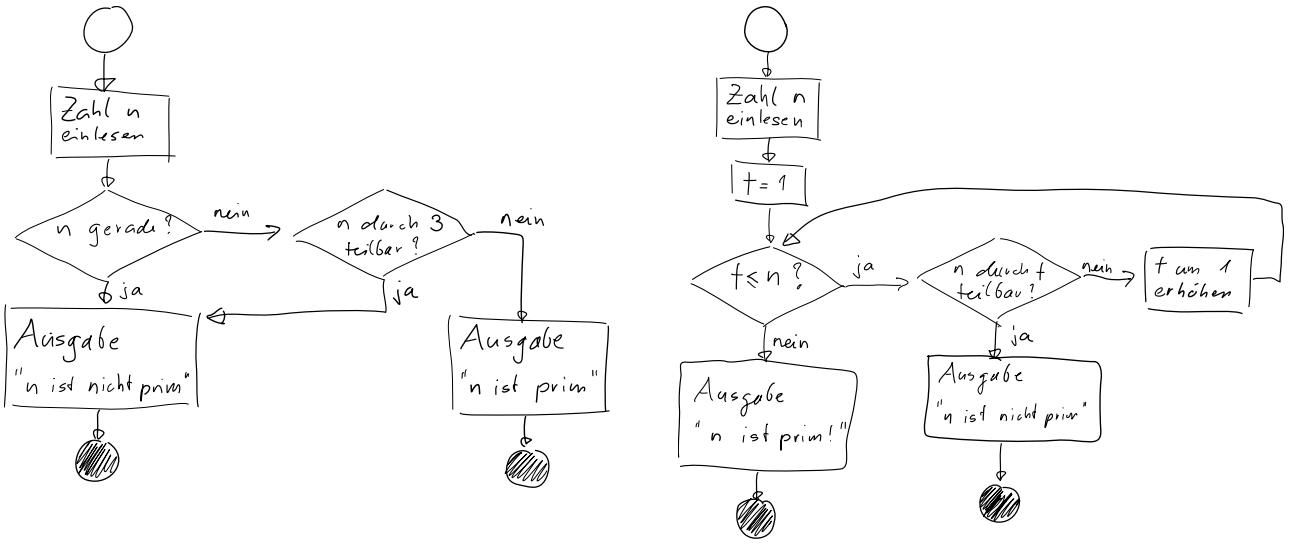
In obigen Programmen mussten diverse Dinge gespeichert werden, wie z.B. das erfragte Alter im Kino, das gewählte Getränk oder der bereits eingeworfene Betrag. Diese Grössen wurden im Programm (bzw. Flussdiagramm) als Wort genannt (z.B. «Alter» oder «Betrag»).

Diese Namen nennen wir **Variablen**. Diese speichern Zahlen, Text oder beliebige komplexere Objekte (wie z.B. Koordinaten, Bilder, etc.).

Variablen, wie ihr Name schon impliziert, können sich während des Programmablaufs ändern (z.B. wird der eingeworfene Betrag nach dem Einwurf einer Münze nachgeführt).

⚒ Aufgabe A6

Folgende Programme sollen entscheiden, ob eine gegebene natürliche Zahl n eine Primzahl ist oder nicht. Leider sind sie fehlerhaft. Finden Sie jeweils den «ersten» Fehler, den das Programm macht. Für welche n «funktioniert» das Programm bereits korrekt?



Entwerfen Sie am Schluss ein korrektes Flussdiagramm.

1.4 Programme als reiner Text

Computerprogramme werden meist als reiner Text geschrieben, wobei Leerschläge und Zeilenumbrüche die einzigen formatierenden Eingaben sind.

Merke 1.4.1 Reine Text-Dateien

Text-Dateien haben einige gewichtige Vorteile:

Menschenlesbar Die Dateien können von Menschen gelesen, verstanden und verändert werden.

Maschinell lesbar Problemlos mit verschiedensten Programmen les- und bearbeitbar.

Ein Nachteil ist, dass Text-Dateien oft mehr Speicherplatz benötigen als z.B. eine Binäre Codierung. Dieser Nachteil kann aber durch den gebrauch von Kompressionsalgorithmen minimiert werden.

Fun fact: Dieses Dokument wurde auch aus einer reinen Text-Datei generiert (mit Hilfe von LaTeX). Z.B. sieht die obige Box im Original wie folgt aus:

```
\begin{merke}[Reine Text-Dateien]
```

```
        Text-Dateien haben einige gewichtige Vorteile:
        \begin{description}
                \item[Menschenlesbar] Die Dateien können von Menschen gelesen,
                        verstanden und verändert werden.
                \item[Maschinell lesbar] Problemlos mit verschiedensten Programmen les-
                        und bearbeitbar.
        \end{description}
        Ein Nachteil ist, dass Text-Dateien oft mehr Speicherplatz benötigen als
        z.B. eine Binäre Codierung. Dieser Nachteil kann aber durch den gebrauch von
        Kompressionsalgorithmen minimiert werden.
\end{merke}
```

Aufgabe A7 Versuchen Sie, den Ablauf des einfachen Getränkeautomats (nur ein Getränk) mit nur Buchstaben, Ziffern und einigen Sonderzeichen (wie Satzzeichen, Operationszeichen, Klammern aufzuschreiben). Als einzige Gestaltungselemente sind Leerschläge und Zeilenumbrüche erlaubt.

Welche Probleme treten dabei auf und wie lösen Sie diese?

1.5 Python Konventionen

Merke 1.5.1 Python Konventionen

- Aufeinander folgende Anweisungen werden gleich eingerückt untereinander geschrieben.
- Verzweigungen werden wie folgt notiert (wobei der «sonst» Teil auch weggelassen werden kann). Man beachte jeweils den Doppelpunkt am Ende der Zeilen mit «Falls» und «Sonst».

```
Falls Bedingung:
   tu dies
   und das
Sonst:
   tu jenes
   und anderes
Tu das danach auf jeden Fall
```

- Wiederholungen werden wie folgt notiert (ebenfalls ein Doppelpunkt am Ende der Zeile mit «Wiederhole»).

```
Wiederhole solange Bedingung:
   tu was wiederholt
   tu noch etwas mehr wiederholen
Tu das nach den Wiederholungen nur einmal
```

Aufgabe A8 Stellen Sie folgende Programme als Flussdiagramme dar.

Programm 1

```
Lese eine natuerliche Zahl n ein.
Speichere 1 in i
Speicher 0 in s
Wiederhole solange i kleiner gleich n:
    Berechne s+i und speichere das Resultat in s
    Erhoehe i um 1
Gebe s aus
```

Programm 2

```
Lese eine natuerliche Zahl n ein.
Wiederhole solange n ungleich 1:
    Falls n gerade:
        Berechne n/2 und speichere das Resultat in n
    Sonst:
        Berechne 3*n+1 und speichere das Resultat in n
    Gebe n aus
```

Was produzieren die Programme für verschiedene Eingaben der Zahl n?

Merke 1.5.2 Python Konventionen

- Zuweisungen zu Variablen (d.h. ein Wert speichern in einer Variablen) werden mit **einem Gleichheitszeichen** wie folgt notiert:

```
a=6
b=7*a
```

- Anstatt «Falls» wird Englisch `if` notiert. Aus «Sonst» wird `else`.
- Anstatt «Wiederhole solange» wird Englisch `while` notiert.
- Anstatt «Ausgabe» wird English `print(`*was*`)` geschrieben.
- Auf Gleichheit wird mit **doppeltem Gleichheitszeichen** geprüft:

```
if b==42:
    print("Die Antwort!")
else:
    print("Nicht korrekt")
```

- Kleiner, Kleiner gleich, Grösser gleich und grösser werden mit `<, <=, >= und >` notiert.
- Ungleich ($\neq$) wird mit `!=` notiert.

```
if x!=0:
    print("x ist nicht Null")
```

Aufgabe A9 Schreiben Sie die Programme von Aufgabe A8 mit den neuen Python Konventionen.

1.6 Lösungen

Hinweise zu den Symbolen:

⚒ Diese Aufgaben könnten (mit kleinen Anpassungen) an einer Prüfung vorkommen. Für die Prüfungsvorbereitung gilt: “If you want to nail it, you’ll need it”.

Diese Aufgaben sind wichtig, um das Verständnis des Prüfungsstoffs zu vertiefen. Die Aufgaben sind in der Form aber eher nicht geeignet für eine Prüfung (zu grosser Umfang, nötige «Tricks», zu offene Aufgabenstellung, etc.). **Teile solcher Aufgaben können aber durchaus in einer Prüfung vorkommen!**.

Diese Aufgaben sind dazu da, über den Tellerrand hinaus zu schauen und/oder die Theorie in einen grösseren Kontext zu stellen.

⚒ Lösung zu Aufgabe A1 ex-schuelerroboter

⚒ Lösung zu Aufgabe A2 ex-steuerung-oder-regelung

a) Liftposition: Gesteuert, z.B. über mechanische Schalter auf der Liftschiene.

b) Lifttüren: Geregelt, die sollen nicht schliessen, wenn etwas im Weg ist.

c) Heizung: Geregelt (Thermostat).

d) Herdplatte: Gesteuert (normalerweise wird einfach eine bestimmte Heizleistung eingestellt).

e) Kühlschrank: Geregelt (Thermostat).

f) Wohnzimmerbeleuchtung: Meistens gesteuert, über einfache Lichtschalter. Kann heute aber auch geregelt sein (Bewegungsmelder).

⚒ Lösung zu Aufgabe A3 ex-kino-einlass

⚒ Lösung zu Aufgabe A4 ex-kino-drei-filme

⚒ Lösung zu Aufgabe A5 ex-getraenke-automat

⚒ Lösung zu Aufgabe A6 ex-logische-fehler-in-flussdiagrammen

⚒ Lösung zu Aufgabe A7 ex-getraenkeautomat-als-text

⚒ Lösung zu Aufgabe A8 ex-pseudocode-zu-flussdiagramm

⚒ Lösung zu Aufgabe A9 ex-pseudocode-pythonisieren

Programm 1

```
Lese eine natuerliche Zahl n ein.
i=1
s=0
while i<=n:
    s=s+i
    i=i+1
print(s)
```

Programm 2

```
Lese eine natuerliche Zahl n ein.
while n!=1:
    if n gerade:
        n = n/2
    else:
        n = 3*n+1
    print(n)
```