



---

Wintersemester 2017/2018

---

## Lineare und kombinatorische Optimierung

### Übungsblatt 1

#### Gruppenübungen

**Aufgabe 1.** (Konvexe Mengen)

(0 Punkte)

Eine Menge  $K \subseteq \mathbb{R}^n$  heißt konvex, wenn

$$\forall x, y \in K \forall t \in [0, 1] : tx + (1 - t)y \in K$$

gilt. Es seien  $m, n \in \mathbb{N}$ . Beweisen oder widerlegen Sie:

- i) Für alle nicht-leeren konvexen Mengen  $K_1, K_2 \subset \mathbb{R}^n$  ist  $K_1 \cup K_2 \subset \mathbb{R}^n$  konvex.
- ii) Für alle nicht-leeren Mengen  $K \subset \mathbb{R}^m$  und  $L \subset \mathbb{R}^n$  ist  $K \times L \subset \mathbb{R}^{m+n}$  konvex genau dann, wenn  $K \subset \mathbb{R}^m$  und  $L \subset \mathbb{R}^n$  konvex sind.

**Lösung:**

- i) Die Aussage ist falsch. Sei etwa  $K_1 = \{0\}$  der Nullvektor und  $K_2 = \{e_1\}$  der erste Einheitsvektor. Dann sind  $K_1$  und  $K_2$  offensichtlich konvex. Der Punkt  $\frac{1}{2}e_1 = \frac{1}{2}0 + (1 - \frac{1}{2})e_1 \notin K_1 \cup K_2 = \{0, e_1\}$ .
- ii) Die Aussage ist richtig. Seien dazu zunächst einmal  $K, L$  konvex und  $(x, x'), (y, y') \in K \times L$  mit  $x, y \in K$  und  $x', y' \in L$ , sowie  $t \in [0, 1]$ . Da  $K$  und  $L$  konvex sind, gilt  $tx + (1 - t)y \in K$ ,  $tx' + (1 - t)y' \in L$  und somit auch  $t(x, x') + (1 - t)(y, y') \in K \times L$ .

Sei nun umgekehrt  $K \times L$  konvex und  $x, x' \in K$  sowie  $t \in [0, 1]$ . Da  $L$  nicht leer ist, gibt es  $y \in L$ . Da  $K \times L$  konvex ist, ist  $t(x, y) + (1 - t)(x', y) = (tx + (1 - t)x', y) \in K \times L$  und somit  $tx + (1 - t)x' \in K$ . Also ist  $K$  konvex. Völlig analog folgt  $L$  ist konvex.

**Aufgabe 2.** (Modellierung)

(0 Punkte)

Ein Erzeuger von Tierfutter produziert ein Gemisch aus drei Bestandteilen: zwei nährstoffreiche Bestandteile und ein Füllmittel. Ein Kilogramm Futter muss einen Minimalgehalt an Nährstoffen enthalten:

Nährstoff	A	B	C	D
Gramm	90	50	20	2

Die nährstoffreichen Bestandteile setzen sich wie folgt zusammen:

	A	B	C	D	Kosten/kg
Bestandteil 1 in g kg <sup>-1</sup>	100	80	40	10	40
Bestandteil 2 in g kg <sup>-1</sup>	200	150	20	–	60

Das Futtermisch soll so erzeugt werden, dass die Kosten möglichst gering sind. Formulieren Sie dies als lineares Optimierungsproblem. Skizzieren Sie die zulässige Menge. Ist sie konvex?

**Lösung:**

Sei

- $x_1$  : kg Bestandteil 1 in einem kg Futtermittel,
- $x_2$  : kg Bestandteil 2 in einem kg Futtermittel.

Mit diesen Variablen lautet das Optimierungsproblem:

$$\begin{aligned}
 \min \quad & 40x_1 + 60x_2 \\
 \text{s.t.} \quad & 100x_1 + 200x_2 \geq 90 \\
 & 80x_1 + 150x_2 \geq 50 \\
 & 40x_1 + 20x_2 \geq 20 \\
 & 10x_1 \geq 2 \\
 & x_1 + x_2 \leq 1 \\
 & x_1, x_2 \geq 0
 \end{aligned}$$

Die zulässige Menge dieses Problems ist durch ein System von linearen Ungleichungen beschrieben und daher konvex. (Skizze siehe Abbildung 1)

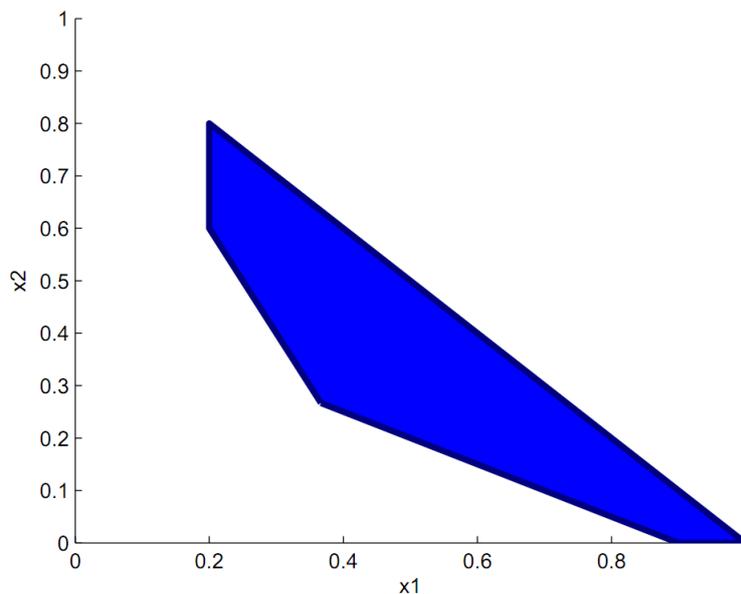


Abbildung 1: Die zulässige Menge

**Aufgabe 3.** (Konvexe Funktionen)

(0 Punkte)

Sei  $f : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$  eine Funktion und  $K \subseteq D$  eine konvexe Menge. Die Funktion  $f$  heißt konvex auf  $K$ , falls

$$x, y \in K, t \in [0, 1] \implies f(tx + (1 - t)y) \leq tf(x) + (1 - t)f(y)$$

gilt.

Für  $i = 1, \dots, r$  seien  $a_i > 0$  und  $f_i : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$  konvexe Funktionen auf einer konvexen Menge  $K \subseteq D$ . Zeigen Sie, dass die Funktion

$$f(x) = \sum_{i=1}^r a_i f_i(x), \quad x \in D,$$

auf  $K$  konvex ist.

**Lösung:**

Seien  $x, y \in K$  und  $t \in [0, 1]$ . Jedes  $f_i$  ist nach Voraussetzung eine konvexe Funktion. Multipliziert man diese mit  $a_i > 0$ , so erhält man die Ungleichung

$$a_i f_i(tx + (1-t)y) \leq t a_i f_i(x) + (1-t) a_i f_i(y).$$

Summation dieser Ungleichung über alle  $i = 1, \dots, r$  liefert

$$\sum_{i=1}^r a_i f_i(tx + (1-t)y) \leq t \sum_{i=1}^r a_i f_i(x) + (1-t) \sum_{i=1}^r a_i f_i(y).$$

Was mit der Definition von  $f$  die folgende gewünschte Beziehung liefert:

$$f(tx + (1-t)y) \leq t f(x) + (1-t) f(y).$$

### Hausübungen

**Aufgabe 4.** (Modellierung)

(4 Punkte)

Eine Nahrungsmittelfirma stellt aus Nüssen, Haferflocken und Rosinen drei Sorten Müsli ( $A, B, C$ ) her. Die Mischungsverhältnisse sind wie folgt gegeben:

	A	B	C
Nüsse	1	3	2
Haferflocken	4	1	3
Rosinen	3	4	3

Das heißt, eine Einheit Müsli  $A$  enthält 1 Einheit Nüsse, 4 Einheiten Haferflocken und 3 Einheiten Rosinen, usw. Beim Verkauf einer Einheit Müsli  $A$  erzielt die Firma einen Gewinn von 5€, der Verkauf von  $B$  bringt 4€ und der Verkauf von  $C$  3€ Gewinn. Die Firma kann maximal 5000 Einheiten Nüsse, 11 000 Einheiten Haferflocken und 8000 Einheiten Rosinen beschaffen. Aufgrund der Nachfrage muss die Firma mindestens drei mal so viele Einheiten von Müsli  $C$  wie von  $A$  herstellen. Ein Produktionsplan mit maximalem Gewinn soll bestimmt werden.

Formulieren Sie das Problem als lineares Optimierungsproblem.

**Lösung:**

Seien  $A, B, C$  die hergestellten Einheiten von Müsli  $A, B$  bzw.  $C$ . Wir stellen folgendes Optimierungsproblem auf:

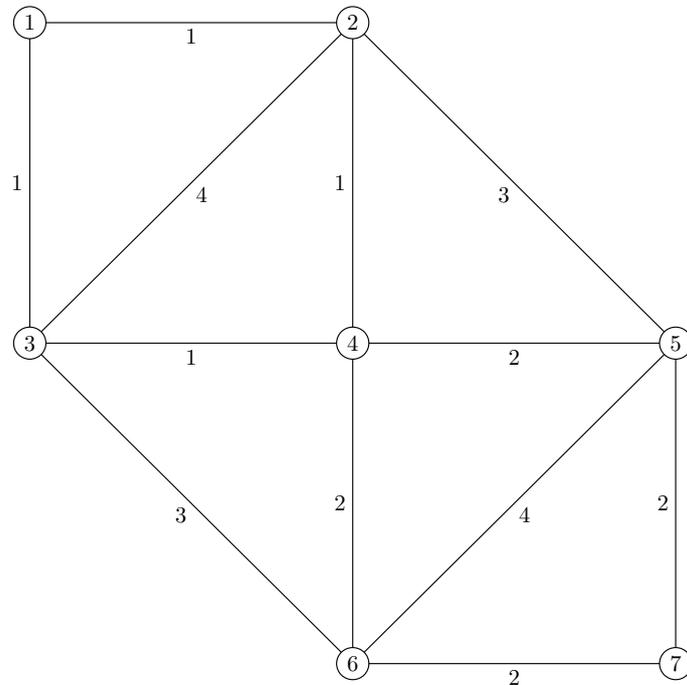
$$\begin{aligned} \max \quad & 5A + 4B + 3C \\ & A + 3B + 2C \leq 5000 \\ & 4A + B + 3C \leq 11000 \\ & 3A + 4B + 3C \leq 8000 \\ & 3A - C \leq 0 \\ & A, B, C \geq 0 \end{aligned}$$

Wie man ein derartiges lineares Optimierungsproblem löst, werden Sie im Verlauf dieser Vorlesung erfahren.

**Aufgabe 5.** (Eigenschaften von Graphen)

(1+1+1+1+2 Punkte)

Gegeben sei der folgende Graph:



1. Geben Sie den dualen Graphen an.
2. Besitzt der Graph eine Eulertour (Eulerkreis) (mit Begründung)?
3. Besitzt der Graph eine Hamiltontour (Hamiltonkreis) (mit Begründung)?
4. Geben Sie die Kodierungslänge des Graphen an.
5. Erstellen Sie für den Graphen eine Adjazenzmatrix und eine Adjazenzliste (Betrachtung ohne Kantengewichte).

**Lösung:**

1. Siehe Abbildung 2.

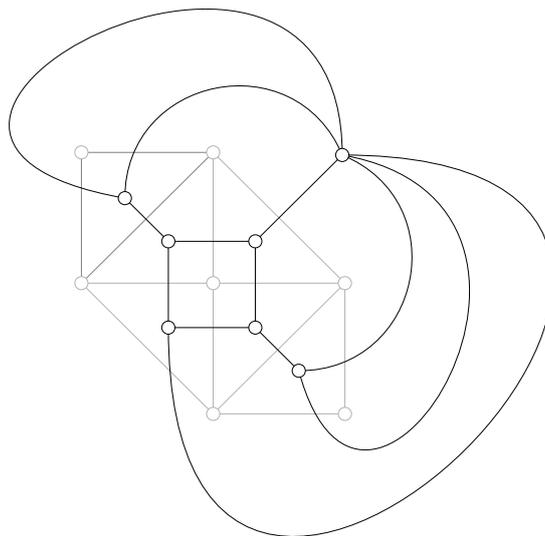


Abbildung 2: Der duale Graph

2. Der Graph besitzt eine Eulertour (Eulerkreis), da alle Knoten einen geraden Grad haben (vgl. Aufgabe 6).
3. Der Graph besitzt eine Hamiltontour (Hamiltonkreis), siehe Abbildung 3.

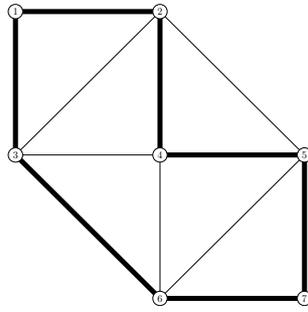


Abbildung 3: Eine Hamiltontour

4.

$$\begin{aligned}
 \langle G_1 \rangle &= |V| + |E| + \sum_{e \in E} \langle c_e \rangle \\
 &= 7 + 12 + 4 \cdot \langle 1 \rangle + 4 \cdot \langle 2 \rangle + 2 \cdot \langle 3 \rangle + 2 \cdot \langle 4 \rangle \\
 &= 19 + 4 \cdot (\lceil \log_2(|1| + 1) \rceil + 1) + 4 \cdot (\lceil \log_2(|2| + 1) \rceil + 1) \\
 &\quad + 2 \cdot (\lceil \log_2(|3| + 1) \rceil + 1) + 2 \cdot (\lceil \log_2(|4| + 1) \rceil + 1) \\
 &= 19 + 4 \cdot 2 + 4 \cdot 3 + 2 \cdot 3 + 2 \cdot 4 \\
 &= 53.
 \end{aligned}$$

$$5. M = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Für die Adjazenzliste haben wir in der Vorlesung folgende Informationen zusammengestellt:

- Die Anzahl der Knoten
- Die Anzahl der Kanten
- Für jeden Knoten den Grad und die inzidenten Kanten

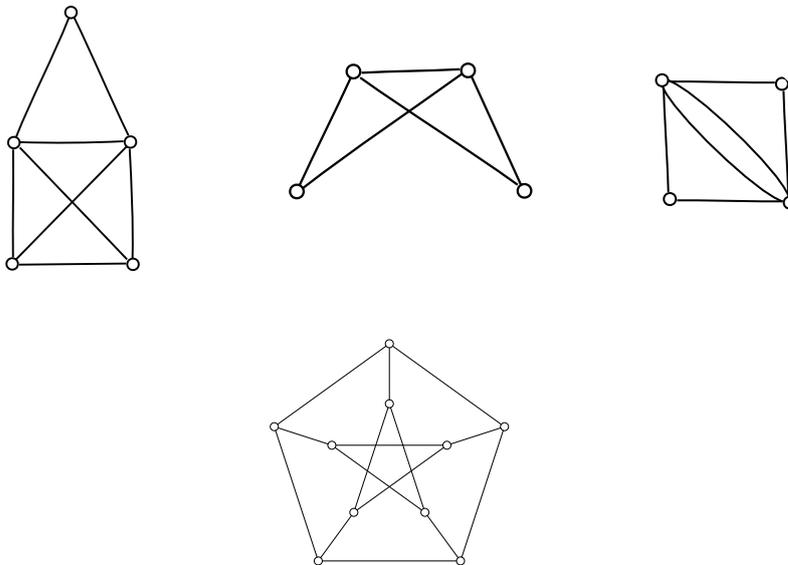
Dies sieht für den gegebenen Graphen folgendermaßen aus:

- 7
- 12
- 1: 2; {1, 2}; {1, 3}
- 2: 4; {1, 2}; {2, 3}; {2, 4}; {2, 5}
- 3: 4; {1, 3}; {2, 3}; {3, 4}; {3, 6}
- 4: 4; {2, 4}; {3, 4}; {4, 5}; {4, 6}
- 5: 4; {2, 5}; {4, 5}; {5, 6}; {5, 7}
- 6: 4; {3, 6}; {4, 6}; {5, 6}; {6, 7}
- 7: 2; {5, 7}; {6, 7}

**Aufgabe 6.** (Eulersche Graphen)

(2+4 Punkte)

1. Welche der Graphen in der Abbildung sind eulersch?



2. Ein Graph heißt zusammenhängend, wenn je zwei Knoten durch einen Weg verbunden werden können. Sei nun  $G = (V, E)$  ein zusammenhängender Graph. Zeigen Sie, dass  $G$  genau dann eulersch ist, falls jeder Knoten  $v \in V$  geraden Grad hat.

*Hinweis:* Benutzen Sie für die Rückrichtung Induktion über die Kantenanzahl und folgendes Lemma:

*Ein Graph, in dem der Grad jedes Knotens mindestens zwei beträgt, enthält einen Kreis, d. h., einen geschlossenen Weg.*

### Lösung:

1. Nein, nein, ja, nein. Um das zu beweisen, zählt man einfach den Knotengrad (vgl. Teilaufgabe 2).
2. ( $\Rightarrow$ ): Sei  $G$  ein zusammenhängender und eulerscher Graph und  $C$  eine Eulertour in  $G$ . Wir wollen zeigen, dass jeder Knoten einen geraden Grad hat. Sei also  $v$  ein beliebiger Knoten in  $V$ . Da die Eulertour jede Kante von  $G$  einschließt, ist  $v$  ein Knoten der Tour.  $v$  wird jedes Mal, wenn er auf der Tour  $C$  durchlaufen wird, von unterschiedlichen Kanten erreicht und verlassen, da jede Kante genau einmal in der Tour vorkommt. Somit wird der Grad von  $v$  jedes Mal um 2 erhöht. Also ist  $\deg(v)$  gerade.

( $\Leftarrow$ ): Sei  $G = (V, E)$  und der Grad von jedem Knoten gerade. Wir zeigen per Induktion über die Kantenanzahl, dass  $G$  eine Eulertour besitzt.

*Induktionsanfang:* Sei  $|E| = 0$ . Da  $G$  zusammenhängend ist, besitzt  $G$  nur einen Knoten. Ein Graph mit nur einem Knoten ist eulersch.

*Induktionsannahme:* Ein Graph mit  $e < |E|$  Kanten und geradem Knotengrad für alle Knoten besitzt eine Eulertour.

*Induktionsschritt:* Wenn der Graph  $G = (V, E)$  Kanten hat, dann gibt es keine isolierten Knoten mit Grad 0, da  $G$  zusammenhängend ist. Da der Knotengrad für alle Knoten gerade ist, gilt daher  $\deg(v) \geq 2$  für alle  $v \in V$ . Nach dem Lemma gibt es also einen Kreis  $C$  in  $G$ . Enthält  $C$  bereits jede Kante von  $G$ , haben wir damit die gesuchte Eulertour. Andernfalls definieren wir einen neuen Graphen  $H := (V, E \setminus E(C))$ , der u.U. nicht zusammenhängend ist. Die Knotengrade in  $H$  sind immer noch gerade, da der Grad jedes mit  $C$  gemeinsamen Knotens genau um 2 reduziert wurde. Nach Induktionsannahme ist jede Zusammenhangskomponente von  $H$  eulersch. Außerdem hat jede Zusammenhangskomponente mindestens einen Knoten mit  $C$  gemein. Wir konstruieren eine Eulertour wie folgt:

Wir starten mit einem beliebigen Knoten in  $C$ .

Laufe solange entlang der Kanten in  $C$ , bis man auf eine Komponente von  $H$  stößt.

Laufe nun entlang der Eulertour in der Komponente.

Am Ende sind wir wieder dort, wo wir  $C$  verlassen haben.

Setze dies fort, bis alle Kanten in  $C$  und alle Eulertouren in allen Komponenten von  $H$  abgelaufen sind.

Da die Kanten von  $G$  genau die Kanten von  $H$  und  $C$  sind, erhalten wir eine Eulertour in  $G$ .

**Aufgabe 7.** (Wege und Kreise)

(2 Punkte)

Sei  $\tau(G) := \min\{\deg(v) \mid v \in V\}$ , d.h. der Minimalgrad eines Knoten in  $G = (V, E)$ .

Beweisen Sie, dass jeder einfache Graph  $G$  einen Weg der Länge  $\tau(G)$  (d.h. Anzahl Kanten) und einen Kreis der Länge mindestens  $\tau(G) + 1$  (für  $\tau(G) \geq 2$ ) enthält.

**Lösung:**

Es sei  $x_0, \dots, x_k$  ein längster Weg in  $G$ . Alle Nachbarn von  $x_k$  in  $G$  liegen dann auf diesem Weg (siehe Abb. 4). Es folgt  $k \geq \deg(x_k) \geq \tau(G)$ . Ist  $i < k$  minimal mit  $x_i x_k \in E$ , so ist  $x_i \dots x_k x_i$  ein Kreis der Länge mindestens  $\tau(G) + 1$ .

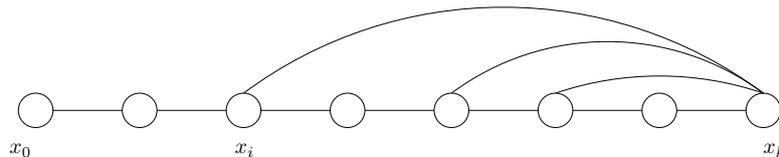


Abbildung 4: Ein längster Weg  $x_0 \dots x_k$ , und die Nachbarn von  $x_k$

**Aufgabe 8.** (Programmieraufgabe: PyCharm, grundlegende Sprachelemente, Debugger) (3 Punkte)

In dieser Aufgabe werden Sie ein paar einfache Methoden in Python mit PyCharm implementieren, sie testen und sich mit dem Debug-Modus von PyCharm vertraut machen. Ein Installationstutorial für Python and PyCharm finden Sie im StudOn unter [Installationstutorial\\_Python.pdf](#).

## 1. Der Start mit PyCharm:

- Öffnen Sie PyCharm und klicken Sie auf **Create New Project** bzw. **File -> New Project**.
- Nennen Sie das Projekt beispielsweise "LKO" und achten Sie darauf, dass als Interpreter `Anaconda3\python.exe` ausgewählt ist. Klicken Sie dann auf **Create**.
- Laden Sie die Datei `ProgAufgabe01.py` aus StudOn herunter und fügen Sie sie in das Projekt ein, beispielsweise per drag-and-drop. Alternativ können Sie auch per Rechtsklick auf den Projektordner und dann mit **New -> Python File** eine neue leere Python-Datei erstellen und den Inhalt der Datei XY in diese kopieren.

## 2. Das Implementieren:

- Ergänzen Sie die Methode `sign(x)`. Diese soll das Signum einer gegebenen Zahl `x` berechnen.
- Ergänzen Sie die Methode `isVectorPositive(v)`. Diese gibt `True` zurück, falls alle Einträge des gegebenen Vektors `v` echt positiv sind, sonst `false`.
- Ergänzen Sie die Methode `sumUp(v)`. Diese addiert alle Einträge des Vektors `v` auf und gibt das Ergebnis zurück.
- Nun können Sie ihre Methoden wie folgt testen: Klicken Sie oben in der Leiste auf **Run -> Run...** und dann auf `ProgAufgabe01`. Nun wird der Code ausgeführt, der unterhalb der von ihnen implementierten Methoden steht. Dieser testet Ihre Methoden.

## 3. Zuletzt testen Sie noch den sog. Debugger von Python. Dieser ermöglicht es Ihnen, den Code stückweise auszuführen und dabei die Werte der einzelnen Variablen auszulesen, was sehr beim Fehler finden hilft:

- Links neben dem Programmierfenster ist ein senkrechter, grauer Balken, in dem die Zeilennummern stehen sollten (wenn nicht kann man die Zeilennummern mit Rechtsklick auf den Balken zuschalten). Wenn Sie rechts neben eine Zeilennummer linksklicken, erscheint dort ein roter Kreis (geht nicht bei Zeilen, in denen kein Code steht).

- Starten Sie ihren Code mittels `Run -> Debug...` im Debug-Modus. Das Programm stoppt nun, sobald eine Zeile mit einem roten Kreis erreicht wird. Mit den blauen Pfeilen im linken unteren Teil des Bildschirms können Sie ihren Code nun Stückweise ausführen. Im Fenster `Variables` werden Ihnen aktuellen Werte der Variablen angezeigt.

**Lösung:**

Die Musterlösung der Programmieraufgabe finden Sie im StudOn in der Datei `ProgAufgabe01_Loesung.py`.

Da am 01.11.2017 Feiertag ist, geben Sie die Hausübungen bitte am 02.11.2017 in der Vorlesung ab.