

LU10.A03 - Einleser erweitern

Ausgangslage

Sie haben in der Aufgabe [LU10.A02](#) eine `input_reader`-Modul erstellt. Damit können Sie jetzt `int` und `float` Zahlen einlesen, und dabei sicherstellen, dass die eingegebenen Werte wirklich einem `int` respektive einem `float` entsprechen.

In dieser Aufgabe erweitern Sie diese Einleser, damit den zwei Funktionen `read_int(text)` und `read_float(text)` neu auch Grenzwerte mitgegeben werden können. Damit könnten Sie dann zum Beispiel auch sicherstellen, dass `note = input_reader.read_float('Bitte eine Schulnote eingeben', lower_bound=1, upper_bound=6)` nur Werte zwischen `1.0` und `6.0` akzeptiert.

Aufgabe

Teilaufgabe 1

Nehmen Sie die Github-Classroom Aufgabe an und clonen Sie das Repository in ihre Entwicklungsumgebung.

Studieren Sie die zwei Python Files `input_reader.py` und `main.py`.

- Sehen Sie im File `main.py` an welcher stelle die Grenzwerte mitgegeben werden? Vielleicht erinnern Sie sich an die [LU09](#) wo wir die benannten Argumente kennengelernt haben.
- Sehen Sie im File `input_reader.py` das wir die Parameter für die Funktion noch nicht mit `lower_bound` und `upper_bound` ergänzt haben? In der [LU09](#) haben wir auch kennengelernt wie sie Default-Werte für Parameter definieren. Dieses Wissen werden wir hier jetzt benötigen.

Teilauftrag 2

Ergänzen Sie die Parameter `lower_bound` und `upper_bound` in der Funktionsdefinition von `read_float` und `read_int`. Damit wir `lower_bound` und `upper_bound` nicht jedes mal angeben müssen, sondern nur dann, wenn wir auch Grenzwerte brauchen, überlegen Sie sich sinnvolle default-Werte für `lower_bound` und `upper_bound` und weisen Sie diese zu. Die Theorie dazu, wie Sie default-Werte zuweisen finden Sie in [LU09](#).



Plus und Minus Unendlich wären vielleicht ganz gute default-Werte? https://www.w3schools.com/python/ref_math_inf.asp

Oder vielleicht einfach `None`, also keiner? Und dann überprüfen ob `if lower_bound is not None and num < lower_bound:`

Entscheiden Sie selbst.

Teilauftrag 3

Überlegen Sie sich, wie Sie die Funktion `read_float(text_to_display, lower_bound = XX, upper_bound = XX)`: anpassen müssen, damit die Überprüfung auf die Grenzwerte funktioniert.

Wichtig Falls der User einen Wert ausserhalb des Grenzwerts eingibt, soll er darauf hingewiesen werden, wo der Grenzwert liegt und ob er oberhalb oder unterhalb davon einen Wert eingeben muss.

```
def read_float(text_to_display, lower_bound = None, upper_bound = None):
    ...

def read_float(text_to_display, lower_bound = float(-inf), upper_bound = float(inf)):
    ...

def read_int(text_to_display, lower_bound = None, upper_bound = None):
    ...

def read_int(text_to_display, lower_bound = int(-inf), upper_bound = int(inf)):
    ...
```

Teilauftrag 4

Importieren Sie ihr `input_reader` Modul in ihrem `main.py` damit das Programm funktioniert.

```
from input_reader import read_int, read_float
def main():
    # do something to show functionality
    mark = read_float('Please enter a mark for your python class> ', lower_bound= 0, upper_bound= 6)
    score_0_to_100 = read_int('Please enter a whole number between 0 and 100> ', lower_bound = 0, upper_bound = 100)
    positive_zahl = read_int('Please enter a positive whole number> ', lower_bound = 1)
    negative_zahl = read_int('Please enter a negative whole number> ', upper_bound = -1)

    print('Your mark is: ', mark)
```

```
print('Your rating is: ', score_0_to_100)
print('Your positive number is: ',
positive_zahl)
print('Your negative number is: ',
negative_zahl)

if __name__ == '__main__':
    main()
```

⇒ *GitHub Repo für externe Besucher*

GitHub Repository <https://github.com/templates-python/m319-lu10-a03-reader-module-extended>

Lernende am BZZ müssen den Link zum GitHub Classroom Assignment verwenden

[M319-LU10](#)



© Kevin Maurizi

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

<https://wiki.bzz.ch/de/modul/m319/learningunits/lu10/aufgaben/einlesererweitert?rev=1750657507>

Last update: **2025/06/23 07:45**

