Aufgabe 90 - Vertiefung Exception auslösen

Ziel

- Sie können eigene Exceptions erzeugen und werfen.
- Sie können mit `try…except` den Fehler auffangen und behandeln

Info

Mit Exceptions können auch eigene Fehlerfälle dokumentiert werden. So kann z.B. der Bereich der Noteneingabe (1.0 bis 6.0) geprüft werden. Im Fehlerfall wird dann ein eigenes Exception-Objekt erzeugt und geworfen mit `raise`.

Exception sind vor allem im Zusammenhang mit der Benutzereingabe bedeutsam. So lässt sich mit einem einfachen Mechanismus die Fehlerbearbeitung an der Stelle auswerten, an der ein Fehler auftritt. Bei graphischen Oberflächen können Fehler soweit delegiert (geworfen) werden, bis sie bei einem zentralen Handler landen, der entsprechende PopUp am Bildschirm anzeigt.

Auftrag

Akzeptieren Sie das Assignment auf GitHub Classroom und klonen Sie das Repository in Ihre Entwicklungsumgebung. Führen Sie dann das Programm aus.

Anzahl der zugefügten Noten überwachen

Ersetzen Sie in der Methode `add_grade()` den Befehl `print(„FEHLER: Zu viele Werte eingegeben\n“)` mit dem nötigen Code, damit im Fehlerfall eine Exception geworfen wird. Lesen Sie dazu falls nötig in der Theorie nach, wie eine `Exception` erzeugt und geworfen wird. Die geworfene Exception soll den selben Kommentar ausgeben wie der `print`-Befehl.
Führen Sie das Programm aus und beachten Sie die Ausgabe im Stdout.

Den Notenwert auf den Bereich 1.0 .. 6.0 überwachen

ACHTUNG: Kommentieren Sie die vorherige Exception aus (mit #) und fügen Sie den Befehla `pass` ein, damit die Codeausführung nicht abgebrochen wird.
Ergänzen Sie die Methode `add_grade()` um eine weitere Fehlerüberprüfung. Falls der Notenwert ausserhalb des gültigen Bereiches (`1.0 <= note <= 6.0`) ist, soll eine Exception geworfen werden, die den Fehlertext „Der Notenwert muss im Bereich 1.0 bis 6.0 liegen“ ausgibt.

```
if elements < self.__MAX_GRADE_COUNT:
  if 1.0 <= grade <= 6.0:
     self.__grades.append(grade)
  else:
     # TODO: Exception erzeugen und werfen
```

Führen Sie das Programm aus und beachten Sie die Ausgabe im Stdout.

Index bei Zugriff auf Liste überwachen

ACHTUNG: Kommentieren Sie die auch diesmal die vorherige Exception aus (mit #) und fügen Sie den Befehla `pass` ein, damit die Codeausführung nicht abgebrochen wird.

Prüfen Sie in der Methode `get_grade()` den Index auf die aktuelle Zahl der Listenelemente. Im Fehlerfall lösen Sie eine Exception aus mit der Meldung „ungültiger Index: ..“.
Führen Sie das Programm aus und beachten Sie die Ausgabe im Stdout.

Abgabe

Geben Sie Ihren Code via Push ins GitHub Repository ab.

M320-LU03

Daniel Fahrni, René Probst, Marcel Suter

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/modul/archiv/m320/learningunits/lu03/aufgaben/lu4-aufgabe_1_v2

Last update: **2026/08/30 14:06**