# LU05.A03: Exceptions auslösen und verarbeiten



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

#### **Ausgangslage**

In der Applikation können eine Anzahl Noten erfasst und ausgelesen werden. Bisher wurde aber noch keine Fehlerbehandlung realisiert. Zum Beispiel stürzt das Programm ab, wenn man eine undefinierte Noten lesen will (falscher Index). Ungültige Noten werden entgegengenommen und wenn zu viele Noten erfasst werden, passiert auch nichts.

Für eine gezielte Fehlerbehandlung definieren wir daher 3 eigene Exceptions:

- ListIndexException: Die Liste hat weniger Elemente als der angegebene Index.
- ListRangeException: Es sollen mehr Elemente als zulässig eingefügt werden.
- ValueRangeException: Der Notenwert ist ungültig (kleiner 1.0 oder grösser 6.0)

#### Vorbereitung

- 1. Akzeptieren Sie das Assignment in GitHub Classroom.
- 2. Klonen Sie das Repository in Ihre Entwicklungsumgebung.
- 3. Führen Sie das Programm aus. Was fällt Ihnen bezüglich dem Programmablauf auf?

### Auftrag 1: Ungültiger Index bei "take\_grade"



Prüfen Sie in der Methode take\_grade(self, index) ob der Index auf ein gültiges Listenelement zeigt. Beachten Sie, dass die maximale Grösse - durch self.\_MAX\_GRADE\_COUNT = 5 festgelegt - nicht dem Index der erfassten Noten entspricht!

- Erstellen Sie die Klasse ListIndexException in der Datei exceptions.py.
  Die Fehlermeldung lautet: "Fehler: Ungültiger Index: ...", wobei ... für den aktuellen Wert des Index steht.
- 2. Passen Sie die Methode take\_grade() so an, dass bei einem ungültigen Index eine ListIndexException erzeugt wird.

Testen Sie die Methode take\_grade() mit den Testfällen test\_list\_index\_ok und test list index exception.

#### Auftrag 2: Zufügen von Noten in "add\_grade" überwachen



Prüfen Sie in der Methode add\_grade(self, grade) ob der Wert einer gültigen Note (1.0 ... 6.0) entspricht und ob nicht zuviele Noten zugefügt werden. Die Grösse der Liste wird - wie oben erwähnt - durch self.\_MAX\_GRADE\_COUNT festgelegt.

- 1. Erstellen Sie die Klassen ListRangeException und ValueRangeException in der Datei exceptions.py.
  - Die Fehlermeldung für ListRangeException lautet : "Fehler: Zu viele Werte eingegeben".
  - Die Fehlermeldung für ValueRangeException lautet: "Fehler: Der Notenwert muss im Bereich 1.0 bis 6.0 liegen. Er beträgt jedoch …", wobei … für den aktuellen Notenwert steht.
- Passen Sie die Methode add\_grade() so an, dass bei einem falschen Notenwert die Exception ValueRangeException und bei zu vielen zugefügten Werten ListRangeException erzeugt werden.
- Testen Sie die Methode add\_grade() mit den Testfällen test\_value\_range\_ok, test\_value\_range\_low, test\_value\_range\_high und test\_number\_too\_large.

## Auftrag 3: Exceptions in "main" überwachen und verarbeiten

In der Methode main() müssen die Exception verarbeitet werden. Überall wo die Methode add\_grade() bzw. take\_grade() aufgerufen werden, müssen Sie dies in einem try / except-Block machen.



Innerhalb eines try-Blocks wird die Programmausführung unterbrochen, sobald eine Exception auftritt, d.h. dass der nachfolgende Code im Block nicht ausgeführt wird. Damit in unserem Fall das Programm weiter läuft, schützen Sie bitte nur die kritischen Bereiche explizit mit try-except (Also nur den einen Aufruf der Methoden add\_grade(...) bzw-take\_garde(...)).

Falls eine Exception auftritt, wird die Standardmeldung der Exception ausgegeben.

https://wiki.bzz.ch/ Printed on 2025/11/20 13:57

- 1. Prüfen Sie in einem ersten Teil des Codes die Anzahl der zugefügten Noten.
- 2. Prüfen Sie dann einen ungültigen Notenwert.
- 3. Prüfen Sie dann den Zugriff auf eine nicht vorhandene Note.
- 4. Führen Sie das Programm aus. Es darf zu keinem Programmabbruch mehr kommen.

#### **Dauer**

45 Minuten

#### **Abgabe**

Mittels Push auf GitHub.

⇒ GitHub Repo für externe Besucher

GitHub Repository https://github.com/templates-python/m320-lu05-a03-raise

Lernende am BZZ müssen den Link zum GitHub Classroom Assignment verwenden

M320-LU05



NO SA René Probst, bearbeitet durch Marcel Suter

From:

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

Permanent link:

https://wiki.bzz.ch/modul/m320\_2024/learningunits/lu05/aufgaben/ausloeser

Last update: 2024/09/04 09:47

