

LU05.L03: Exceptions auslösen und verarbeiten

gradelist.py

```
from exceptions import ValueRangeException, ListRangeException,
ListIndexException

class GradeList:
    """
    Eine Liste von Notenwerte.
    Die Liste ist auf MAX_GRADE_COUNT Elemente begrenzt. Zudem dürfen
    die
    Notenwerte nur im Bereich 1.0 ... 6.0 liegen.
    Beide Zusicherungen werden in der Methode add_grade überprüft.
    """

    def __init__(self):
        """
        Definiert den Range der Liste und erstellt eine leere Liste.
        """
        self._MAX_GRADE_COUNT = 5
        self._grades = []

    def __str__(self):
        """
        Gibt alle Notenwert am Stdout aus.
        """
        r = range(self.current_grade_count)
        output = ''
        for i in r:
            output += f'{i + 1}. Note: {self._grades[i]}\n'
        return output

    def add_grade(self, grade):
        """
        Fügt einen Notenwert der Liste zu.
        Der Wert muss im Bereich 1.0 ... 6.0 liegen.
        Es können maximal MAX_GRADE_COUNT Elemente der Liste zugefügt
        werden.
        :param grade: ein Notenwert
        """
        elements = self.current_grade_count
        if elements < self._MAX_GRADE_COUNT:
            if (1.0 <= grade <= 6.0):
                self._grades.append(grade)
```

```
        else:
            raise ValueErrorException(grade)
    else:
        raise ListRangeException(self._MAX_GRADE_COUNT)

@property
def max_grade_count(self):
    """
    Liefert die maximale Grösse der Liste.
    :return: Grösse der Liste
    """
    return self._MAX_GRADE_COUNT

@property
def current_grade_count(self):
    """
    Liefert die aktuelle Anzahl der in der Liste abgelegten
    Notenwerte.
    :return: Anzahl Notenwerte
    """
    return len(self._grades)

def take_grade(self, index: int):
    """
    Liefert den durch index bezeichneten Notenwert aus der Liste.
    :param index: Position des Notenwertes
    :return: Notenwert
    """
    if index <= self.current_grade_count:
        return self._grades[index]
    else:
        raise ListIndexException(index)

def remove_grade(self, index: int):
    """
    Entfernt an der Stelle index einen Wert.
    :param index: Position die entfernt wird
    """
    self._grades.pop(index)
```

exceptions.py

```
class ListIndexException(Exception):
    def __init__(self, position):
        super().__init__(f'An {position} ist keine Note gespeichert')

class ListRangeException(Exception):
```

```
def __init__(self, size):  
    super().__init__(f'Es können maximal {size} Noten gespeichert  
werden')  
  
class ValueErrorException(Exception):  
    def __init__(self, value):  
        super().__init__(f'Die Note {value} liegt nicht im Bereich 1  
... 6')
```

M320-LU05

✖ René Probst, bearbeitet durch Marcel Suter

From:

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

Permanent link:

https://wiki.bzz.ch/modul/m320_2024/learningunits/lu05/loesungen/ausloesen

Last update: **2024/08/26 13:45**

