

Lösung 4 - Exception auslösen und bearbeiten

```
from builtins import int

class ListIndexException(Exception):
    def __init__(self, index: int):
        super().__init__(f'Fehler: Ungültiger Index: {index}')

class ListRangeException(Exception):
    def __init__(self):
        super().__init__(f'Fehler: Zu viele Werte eingegeben')

class ValueRangeException(Exception):

    def __init__(self, grade: float):
        super().__init__(f'Fehler: Der Notenwert muss im Bereich 1.0 bis 6.0 liegen. Er beträgt jedoch {grade}.')

class GradeList:
    def __init__(self):
        self._MAX_GRADE_COUNT = 5
        self._grades = []

    def add_grade(self, grade: float) -> None:
        if 1.0 <= grade <= 6.0:
            elements = self.current_grade_count
            if elements >= self._MAX_GRADE_COUNT:
                raise ListRangeException()
            print(f'zufügen von Note mit {grade}' )
            self._grades.append(grade)
        else:
            raise ValueRangeException(grade)

    @property
    def max_grade_count(self) -> int:
        return self._MAX_GRADE_COUNT

    @property
    def current_grade_count(self) -> int:
        return len(self._grades)

    def take_grade(self, index: int) -> float:
        if 0 <= index < self.current_grade_count:
            return self._grades[index]
        else:
            raise ListIndexException(index)
```

```
def remove_grade(self, index: int) -> None:
    self._grades.pop(index)

def print(self) -> None:
    r = range(self.current_grade_count)
    for i in r:
        print(f"{i + 1}. Note: {self._grades[i]}")

if __name__ == '__main__':
    demo = GradeList()
    demo.add_grade(4.5)
    demo.add_grade(5.0)
    demo.add_grade(3.5)
    demo.add_grade(4.0)
    demo.add_grade(4.5)
    demo.print()
    # und einen nächsten Wert einfügen, der zu einem Overflow führen wird.
    try:
        print("\nund nun einen weiteren Wert zufügen")
        demo.add_grade(3.5)
    except ListRangeException as lre:
        print(lre)
    demo.print()

    print("\nLösche Wert an 2. Stelle")
    demo.remove_grade(1) # Index beginnt bei 0
    demo.print()

    # und nun einen Wert zufügen der eine ungültige Note darstellt.
    print("\nund nun eine ungültige Note zufügen")
    try:
        demo.add_grade(7.0)
    except ValueError as lre:
        print(lre)
    demo.print()

    # und nun einen Wert an einer Stelle lesen, die es nicht gibt
    print("\nNote an Position 8 lesen ")
    try:
        print(demo.take_grade(8))
    except ListIndexException as lie:
        print(lie)

    print(f"\nListe umfasst zur Zeit {demo.current_grade_count} Noten")
    print(f"Note an 3. Stelle ist {demo.take_grade(2)}")
    print(f"Grösse der Liste beträgt {demo.max_grade_count}\n")
```

```
demo.print()
```

From:

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

Permanent link:

https://wiki.bzz.ch/modul/m320/learningunits/lu03/loesungen/lu4-aufgabe_2

Last update: **2024/03/28 14:07**

