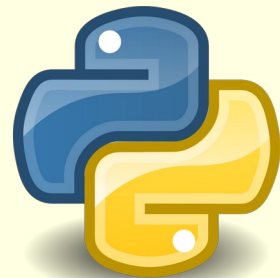




Fehlerbehandlung

Fehlerbehandlung



jeder Fehler eigene Klasse

- abgeleitet von Exception (empfohlen)
- über Exception: BaseException
 - im allgemeinen nicht abgefangen
 - System: KeyboardInterrupt und SystemExit
 - nicht empfohlen

Fehlerbehandlung



Laufzeitfehler abfangen

try...except

```
1
2     try:
3         ...
4     except TypeError as e:
5         ...
6     except:
7         ...
8     else:
9         ...
10    finally:
11        ...
```

Fehlerbehandlung



Laufzeitfehler abfangen

try...except

- except: als letztes
- else: kein Fehler
- finally: immer

```
1
2     try:
3         ...
4     except TypeError as e:
5         ...
6     except:
7         ...
8     else:
9         ...
10    finally:
11        ...
```

Fehlerbehandlung



Fehler selbst definieren

- ableiten von Basisklasse *Exception*
- nach Belieben Attribute und Methoden

Fehlerbehandlung



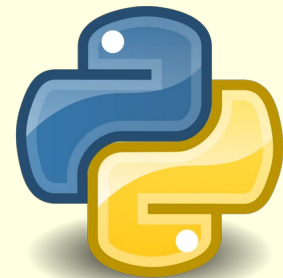
Fehler selbst definieren

- von Exception

```
1 class Wertebereichsfehler(Exception):
2     pass
3
4 def int_check(k):
5     if k < 10 or k > 1000:
6         raise Wertebereichsfehler()
7
8 int_check(22)
9 print("bis hierher")
10 int_check(2345)

bis hierher
...
Wertebereichsfehler
```

Fehlerbehandlung



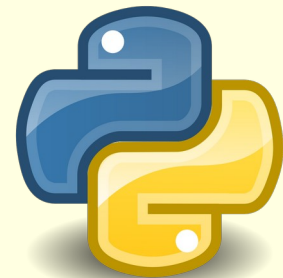
Fehler selbst definieren

- args übergeben
wie bisher

```
1 class Wertebereichsfehler(Exception):
2     pass
3
4 def int_check(k):
5     if k < 10:
6         raise Wertebereichsfehler("Wert zu klein")
7     elif k > 1000:
8         raise Wertebereichsfehler("Wert zu gross")
9
10 int_check(2345)
11
12
```

Wertebereichsfehler: Wert zu gross

Fehlerbehandlung



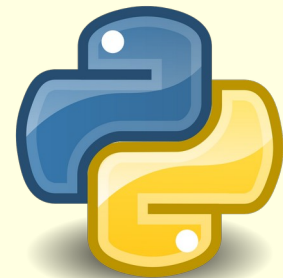
Fehler selbst definieren

- args übergeben
- Ausgabe nur durch `__str__()`

```
1 class Wertebereichsfehler(Exception):
2     def __str__(self):
3         return "Wert außerhalb des Bereichs"
4
5 def int_check(k):
6     if k < 10:
7         raise Wertebereichsfehler("Wert zu klein")
8     elif k > 1000:
9         raise Wertebereichsfehler("Wert zu gross")
```

`Wertebereichsfehler`: Wert außerhalb des Bereichs

Fehlerbehandlung



Fehler selbst definieren

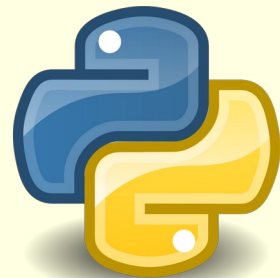
mehr

Argumente

```
1 class Wertebereichsfehler(Exception):
2     def __init__(self, msg, k):
3         self.nachr = msg
4         self.wert = k
5     def __str__(self):
6         return self.nachr + ": " + str(self.wert)
7
8 def int_check(k):
9     if k < 10:
10         raise Wertebereichsfehler("Wert zu klein", k)
11     elif k > 1000:
12         raise Wertebereichsfehler("Wert zu gross", k)
```

Wertebereichsfehler: Wert zu gross: 2345

Fehlerbehandlung



Fehler selbst definieren

except-Block für erste

Oberklasse wird angewendet

```
1 class F(Exception):  
2     pass  
3 class F2(F):  
4     pass  
5  
6 try:  
7     raise F2("Fehler 2!")  
8 except F:  
9     print("F")  
10 except F2:  
11     print("F2")
```

F

Fehlerbehandlung



Fehler selbst definieren

- Ausführung von
Methoden über Instanz

```
1 class MeinFehler(Exception):
2
3     def ausnahmemethode( self, x ):
4         print( " aus der Methode: ", x )
5
6 try:
7     raise MeinFehler
8
9 except MeinFehler as e:
10     e.ausnahmemethode( 77 )
```

aus der Methode: 77

Fehlerbehandlung



Fehler selbst definieren

können auch nicht-Exceptions
auslösen und abfangen

```
1  try:
2      raise list
3
4
5
6  except:
7      print("Halb so wild.")

Halb so wild.
```

Fehlerbehandlung



Fehler selbst definieren

können auch nicht-Exceptions
auslösen und abfangen

- aber nicht in spezifischem
except

```
1 try:
2     raise list
3
4 except list:
5     print("listenspezifisch")
6 except:
7     print("Halb so wild.")
```

TypeError: catching classes that do not inherit from BaseException is not allowed



What is the expected outcome of the following code?

```
1 try:
2     raise Exception
3
4 except BaseException:
5     print("a")
6 except Exception:
7     print("b")
8 except:
9     print("c")
```

- a) b
- b) a
- c) Syntax Error
- d) c



Aufgabe

Erstellen Sie eine Funktion `minus_pos(x,y)`, die

- `x-y` zurückliefert, falls der Wert positiv ist
- für `x-y == 0` soll ein Fehler `NullwertFehler` ausgelöst werden, für negative Ergebnisse ein `NegativwertFehler`; beide sollen von `WertFehler` abgeleitet werden
- fangen sie beide Fehler im Hauptprogramm bei einem Aufruf von `minus_pos()` ab und geben Sie entsprechende Erklärungen aus; Beim `NegativwertFehler` soll die Meldung das unzulässige Ergebnis enthalten.