Wiederholungsstrukturen

Wiederholte Ausführung
von
Anweisungsblöcken

[*Schleifen*]

Wiederholungsstrukturen

- Sammlungsstrukturen und Wiederholungsstrukturen gehören unmittelbar zusammen.
  - wiederhole das Einfügen eines neuen Elements in eine Sammlung
  - wiederhole den Zugriff auf Elemente in einer Sammlung

Wiederholungsstrukturen

- Beispiel Schrankwandklasse
  - Sammlung ist eine Liste

```
self.__schraenke = []
```

  - Wiederholung beim Einfügen (*Zählschleife*)

```
for i in range(anzahl):
    self.__schraenke.append(Schrank(...))
```

  - Wiederholung beim Zugriff (*for each*)

```
for schrank in self.__schraenke:
    path.AddPath(schrank.GibFigur())
```

Wiederholungsstrukturen

- Beispiel Texte
  - Sammlung ist ein String

```
satz = ""
```

  - Wiederholung beim Einfügen geht nicht, da statisch, aber mit Neudefinition

```
while eingabe != ".":
    satz += " " + eingabe
```

  - Wiederholung beim Zugriff

```
for buchstabe in wort:
    print(ord(buchstabe))
```

ord("A") → 65

Wiederholungsstrukturen

- Beispiel Vektoren mit Tupeln
  - Sammlung ist ein Tupel

```
vektor = (2,3,4)
```

  - Wiederholung beim Einfügen geht nicht, da statisch

Neudefinition mit +=

  - Wiederholung beim Zugriff (Zählschleife)

```
laenge = 0
for i in range(len(vektor)):
    laenge += vektor[i] ** 2
laenge = math.sqrt(laenge)
```

Wiederholungsstrukturen

- Beispiel Vektoren mit Tupeln
  - Sammlung ist ein Tupel

```
vektor = (2,3,4)
```

  - **alternative Wiederholung** beim Zugriff

```
laenge = 0
for komponente in vektor:
    laenge += komponente ** 2
laenge = math.sqrt(laenge)
```

Wiederholungsstrukturen

- Beispiel eigene Klasse Warteschlange
  - Sammlungsklasse ist selbst geschrieben und implementiert `__iter__`

```
def __iter__(self):
    '''initialisiert den Iterator'''
    self.aktuell=self.__kopf
    return self
```

und ...

Wiederholungsstrukturen

- Beispiel eigene Klasse Warteschlange
  - ... und `next()`

```
def next(self):
    '''Iterationsschritt'''
    if self.aktuell==None:
        raise StopIteration
    else:
        wert=self.aktuell.GibInhalt()
        self.aktuell=
            self.aktuell.GibNachfolger()
        return wert
```

Wiederholungsstrukturen

- Beispiel eigene Klasse Warteschlange
  - Sammlungsklasse implementiert **`__iter__`** und **`next()`**

```
warteschlange = Warteschlange()

warteschlange.Anstellen('Meier')

...
```

  - **Wiederholung** beim Zugriff

```
for inhalt in warteschlange:
    print ( inhalt )
```

Wiederholungsstrukturen

Schleifen …
Es gibt nicht nur **`for`**

oder
„gut Ding hat auch **`while`**”

aber:
ernsthaft ein Unterschied?

Wiederholungsstrukturen

- Beispiel:
  Suche der Einfügeposition
  eines neuen Elements
  in einer vorsortierten Liste

```
def suchePosition(wort, liste):
    i=0
    while (i < len(liste)) and (element > liste[i]) :
        i+=1
    return i
```

zuerst prüfen!

Wiederholungsstrukturen

- Wann verwendet man … ?
- for
  - Zählschleife mit vorher bekannter Anzahl von Schritten
  - vollständige Iteration über eine Sammlung
- while
  - das Ende der Bearbeitung hängt von der Laufbedingung ab und ist im Prinzip nicht vorher bekannt.