



8. Online-Abgabe

Abzugebende Dateien: `ring.py`

Deadline: Sonntag, 19.12.2011, 20:00 Uhr

Abgabe-URL: `.../~barth/cgi-bin/upload.cgi`

Aufgabe 1. Schreiben Sie eine Python-Datei `ring.py` mit einer Klasse `Ring`. Die Klasse `Ring` implementiert einen Ring von Elementen. Ein Ring von Elementen ist eine unendliche Sequenz aus endlich vielen Elementen, die sich immer wieder wiederholen. Innerhalb der endlich vielen Elemente kann es doppelte Vorkommen des gleichen oder identischen Elements geben. Jeder Ring hat ein aktuelles Element und eine Einfügestelle vor der aktuellen Position. Die Klasse `Ring` soll sich mit einer Sequenz initialisieren lassen. Folgende Methoden sollen zur Verfügung stehen:

- `add(ele)`: Fügt Element `ele` zu einem Ring vor der aktuellen Stelle hinzu.
- `add_sequence(seq)`: Fügt die Elemente einer Sequenz `seq` hintereinander in der Reihenfolge der Sequenz dem Ring hinzu.
- `get_elements()`: Gibt die endlich vielen Elemente eines Rings als Liste ab der aktuellen Stelle zurück.
- `get_current()`: Gibt das aktuelle Element zurück.
- `remove_current()`: Entfernt das aktuelle Element.
- `next()`: Gibt das aktuelle Element zurück, und rückt danach eins weiter.
- `is_empty()`: Ist der Ring leer?
- `clear()`: Macht den Ring leer.
- Ist iterierbar und gibt einen Iterator zurück, der unendlich oft das nächste Element eines Rings, so wie der Ring zur Aufrufzeit existierte, zurück gibt.
- Kann man auf der interaktiven Konsole wie im Beispiel auf der nächsten Seite, mit `print` genauso ausgeben und gibt mit `len` die Anzahl der endlich vielen Elemente aus.
- Der `in` Test funktioniert und man erhält mit `[i]` das `i`-te Element des Rings ab dem aktuellen Element.



Beispielablauf:

```
>>> from ring import Ring
>>> r = Ring([1,2])
>>> r
r[; 1, 2]
>>> r.add(3); r
r[; 1, 2, 3]
>>> r.next(), r.next(), r.next(), r.next()
(1, 2, 3, 1)
>>> r
r[1; 2, 3]
>>> r.add(4); r
r[1, 4; 2, 3]
>>> r.next()
2
>>> print len(r), r
4 r[1, 4, 2; 3]
>>> 6 in r, 2 in r
(False, True)
>>> r.get_current(), r[0], r[1], r[2], r[3], r[4]
(3, 3, 1, 4, 2, 3)
>>> r
r[1, 4, 2; 3]
>>> r.get_elements()
[3, 1, 4, 2]
>>> r.remove_current(); r
r[; 1, 4, 2]
>>> it = iter(r)
>>> it.next(), it.next(), it.next(), it.next(), it.next()
(1, 4, 2, 1, 4)
>>> r
r[; 1, 4, 2]
>>> r.add_sequence((5,6,7)); r
r[; 1, 4, 2, 5, 6, 7]
>>> it.next(), it.next(), it.next(), it.next(), it.next()
(2, 1, 4, 2, 1)
>>> r.is_empty()
False
>>> r.clear(); r.is_empty()
True
```

<https://login3.mi.hs-rm.de/~barth/cgi-bin/upload.cgi>