

Aufgabe zu Queues

Implementieren Sie die angegebenen Methoden einer Klasse `Queue` für Warteschlangen. Eine Warteschlange soll eine unbeschränkte Anzahl von Elementen aufnehmen können. Elemente sollen am Ende der Warteschlange angefügt und am Anfang aus ihr entfernt werden. Sie können davon ausgehen, dass ein Klasse `QueueElement` mit der folgenden Schnittstelle bereits implementiert ist .

```
3  class QueueElement {
4
5      private QueueElement next;
6      private Object contents;
7
8      QueueElement(Object contents) {
9          this.contents = contents;
10     }
11
12     Object getContents() {
13         return contents;
14     }
15
16     QueueElement getNext() {
17         return next;
18     }
19
20     void setNext(QueueElement next) {
21         this.next = next;
22     }
23 }
```

Von der Klasse `Queue` ist folgendes gegeben:

```
3  class Queue {
4      QueueElement first;
5      QueueElement last;
```

- (a) Schreiben Sie eine Methode `void append (Object contents)`, die ein neues Objekt in der Warteschlange einfügt.

```
7      public void append(Object contents) {
8          QueueElement newElement = new QueueElement(contents);
9          if (first == null) {
10              first = newElement;
11              last = newElement;
12          } else {
13              // neues Element hinten anhaengen
14              last.setNext(newElement);
15              // angehaengtes Element ist Letztes
16              last = last.getNext();
17          }
```

- (b) Schreiben Sie eine Methode `Object remove()`, die ein Element aus der Warteschlange entfernt und dessen Inhalt zurückliefert. Berücksichtigen Sie, dass die Warteschlange leer sein könnte.

```

20 public Object remove() {
21     Object temp = null;
22     if (first != null) {
23         // Inhalt des Ersten temporaer speichern
24         temp = first.getContents();
25         // Erstes aus der Schlange nehmen
26         first = first.getNext();
27     }
28     return temp;
29     // Inhalt des geloeschten ausgeben bzw . null
30 }

```

- (c) Schreiben Sie eine Methode `boolean isEmpty()`, die überprüft, ob die Warteschlange leer ist.

```

32 public boolean isEmpty() {
33     return (first == null);
34 }

```