

1. Du kannst mindestens drei Anwendungen der Datenstruktur Queue angeben.
2. Du kannst die für die Datenstruktur typischen Operationen beschreiben:
  - Eine leere Queue erzeugen: `q = Queue()`
  - Ein Element hinzufügen: `q.enqueue(item)`
  - Ein Element entfernen und zurückgeben: `q.dequeue()`
  - Alle Elemente entfernen: `q.clear()`
  - Testen, ob die Queue leer ist: `q.is_empty()`
  - Die Anzahl der Elemente zurückgeben: `q.size()`
3. Du kannst Python-Code nachvollziehen, der die oben angegebenen Operationen auf konkreten Daten anwendet.
4. Du kannst begründen, warum sich Python-Listen mit den Operationen
  - `append(item)` [Einfügen am Listenende] und
  - `pop(0)` [Entfernen am Listenanfang]nicht zur Implementierung der Operationen `enqueue(item)` und `dequeue()` von Stacks eignen.
5. Du kannst in Python einfach verlinkte Listen mit Hilfe der Klasse `Node` bilden. Der Code der Klasse wird gegeben.
6. Du kannst eine einfach verkettete Liste mit einem zusätzlichen Zeiger auf das letzte Element anhand ihres Speicherabbildes graphisch darstellen.
7. Du kannst die *enqueue*- und *dequeue*-Operationen auf einer Queue graphisch darstellen.