

Queues in Python

Jacques Mock Schindler

19.02.2025

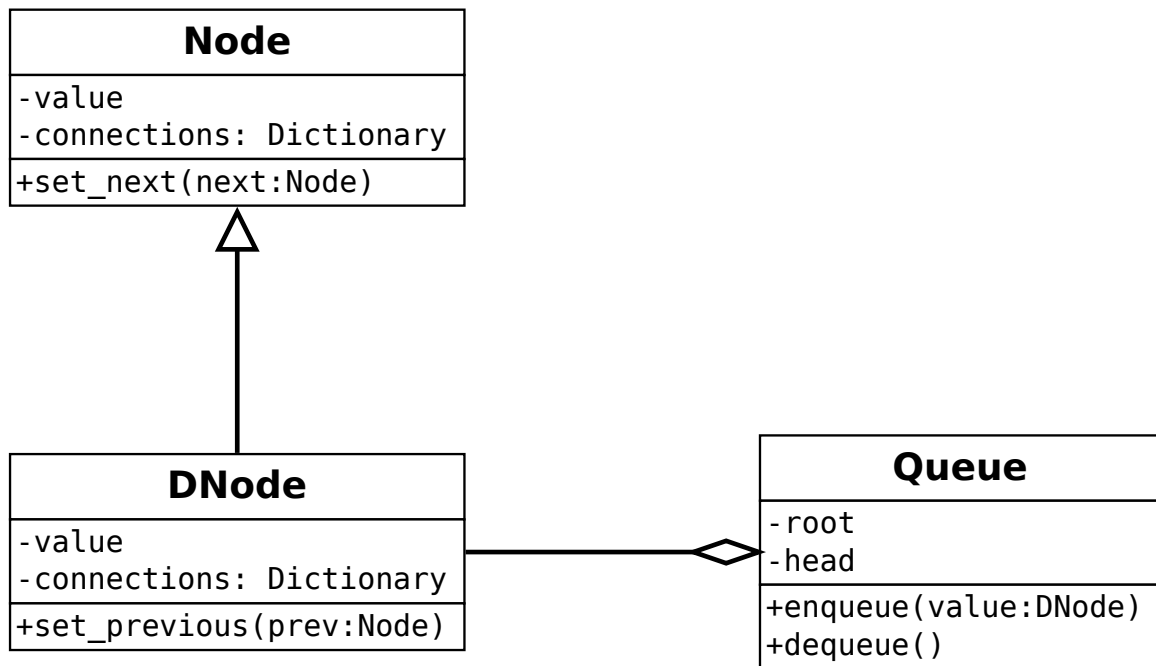
Queues sind Datenstrukturen, welche Daten speichern und grundsätzlich in der Reihenfolge, in der sie abgespeichert worden sind, wieder zurückgeben (First In - First Out, FIFO). Eine Queue ist eine derart fundamentale Datenstruktur, dass Python sie als [Library](#) zur Verfügung stellt.

Um zu verstehen, wie eine Queue funktioniert, geht es im folgenden darum, eine eigene Klasse Queue in Python zu implementieren.

Zu Beginn ist zu überlegen, welche Eigenschaften, die Queue aufweisen muss. Sie muss Daten abspeichern und diese in der gleichen Reihenfolge wieder ausgeben können. Wir brauchen also eine Struktur für die Daten und eine Struktur, welche die Reihenfolge der Speicherung festhält. Die Struktur, welche die Reihenfolge festhält, muss ausserdem in der Lage sein, neue Daten abzuspeichern und bereits abgespeicherte Daten wieder zurückzugeben. Diese Anforderungen können mit Hilfe bereits programmierter Klassen umgesetzt werden. Um die Daten abzuspeichern können wir Nodes verwenden und für die Struktur zum Erhalt der Reihenfolge die Linke List.

In der aktuellen Implementation der Linked List gibt es nur einen Positionsbezug auf das letzte eingefügte Element (`self.root`). Damit die Linked List als Queue verwendet werden kann, muss auch ein Bezug auf das erste eingefügte Element angelegt werden (`self.head`). Für das Erste überhaupt in die Datenstruktur eingefügte Element ist dies kein Problem. Wenn aber weitere Elemente eingefügt oder entfernt werden, dann muss in den einzelnen Nodes nicht nur ein Bezug auf das folgende Element (`self.connections['next']`) sondern auch einer auf das Vorangehende Element (`self.connections['previous']`). Entsprechend müssen die beiden Klassen Node und Linked List angepasst werden.

Das kann umgesetzt werden, in dem basierend auf den bereits existierenden Klassen abgeleitete Klassen implementiert werden. Als UML-Klassendiagramm sieht das folgendermassen aus:



Für die Umsetzung der obigen Ausführungen stehen hier zwei Module ([linked_list.py](#) und [nodes.py](#)) zur Verfügung.