**Beschreibung der Implementierung von Stack und Queue**

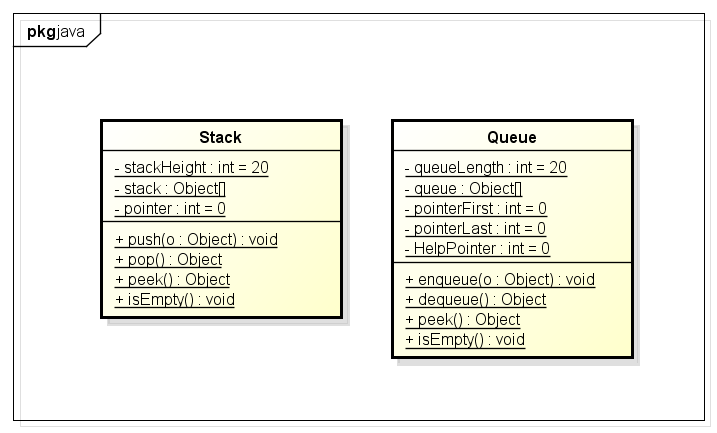
Ein Stack arbeitet nach dem LIFO-Prinzip. Das Element, welches als letztes auf den Stack gelegt wurde, muss auch als erstes wieder herausgenommen werden:

Hierzu wird also nur ein Pointer benötigt, welcher jeweils auf das nächste freie Feld zeigt. Ist der Stack leer, zeigt dieser Pointer auf das erste Feld, ist der Stack voll, zeigt er auf das letzte Feld+1.

Eine Queue hingegen arbeitet nach dem FIFO-Prinzip. Das Element, welches als erstes in die Queue geschrieben wurde, muss auch als erstes wieder herausgenommen werden:

Hierzu werden also zwei Pointer benötigt. Einer zeigt immer auf den Anfang der Queue und der zweite immer auf das Ende. Zeigen die Pointer auf das gleiche Feld, ist die Queue leer. Wenn sich folgender Sachverhalt ergibt, ist die Queue voll:  
AnfangsPointer = EndPointer+1  
Denn in diesem Fall liegt das erste Element im Feld, welches als nächstes beschrieben würde.

Eine Schwierigkeit besteht darin, diesen Sachverhalt festzustellen, sobald die Queue einmal über ihren Rand hinauswachsen soll. Folgende Annahme:

Die Queue besteht aus 3 Elementen (0-2) und ist voll. Anfangspointer= 0 und EndPointer = 2. Nun wird versucht ein weiteres Element hinzuzufügen. Dieses Element würde nun wieder an Position 0 eingefügt werden. Da diese aber bereits belegt ist muss dieser Fall abgefangen werden.  
AnfangsPointer = EndPointer+1 gilt hier allerdings nicht.  
Daher wird eine Hilfsvariable benötigt, welche anzeigt ob der Rand der Queue beim nächsten Schreibvorgang überschritten wird. Ist dies der Fall und der AnfangsPointer zeigt nach wie vor auf 0, ist die Queue voll.

UML-Diagramme: