

Aufgabenblatt A04: Datenstrukturen Queue und Stack

1. Ein *Queue* ist eine Datenstruktur, bei der nur „am Ende“ Elemente eingefügt und nur „am Anfang“ entnommen werden können. Eine Queue beschreibt also so etwas, wie eine Warteschlange an einer Supermarktkasse. Man spricht auch von einem „FiFo, First in - First out“-Speicher.

Implementieren Sie bitte eine *Queue* mit folgenden Methoden!

- 1.1. enqueue: Einfügen eines neuen Elements am Ende.
 - 1.2. dequeue: Entfernen des Elements am Anfang.
 - 1.3. peek: Gibt das Element am Anfang zurück, belässt es aber in der Queue.
2. Ein *Stack* ist eine Datenstruktur, bei der nur „am Ende“ Elemente eingefügt und entnommen werden können. Ein Stack beschreibt also so etwas, wie einen Kartenstapel, einen „LiFo, Last in - First out“-Speicher.

Implementieren Sie bitte einen *Stack* mit folgenden Methoden!

- 2.1. push: Einfügen eines neuen Elements am Ende.
- 2.2. pop: Entfernen des Elements am Ende.
- 2.3. peek: Gibt das Element am Anfang zurück, belässt es aber im Stack.

Treffen Sie für die folgenden Fragen bitte Entscheidungen:

- 2.1. Überlegen Sie sich bitte ob und ggf. welche Parameter die Methoden benötigen und begründen Sie Ihre Entscheidung!
- 2.2. Überlegen Sie sich bitte, was bei *dequeue* und *pop* zurückgegeben werden soll, wenn die *Queue* bzw. der *Stack* leer sind und begründen Sie Ihre Entscheidung!
- 2.3. Überlegen Sie sich bitte, was bei *enqueue* und *push* zurückgegeben werden soll und begründen Sie Ihre Entscheidung!
- 2.4. Denken Sie bitte auch an *to_s* und ggf. weitere wichtige Methoden, falls wir bis zum Freitag, 27.10.2017 in der Vorlesung soweit sind.

3. Schreiben Sie bitte aussagefähige RUnit-Testfälle, um zu zeigen, dass Ihre Implementierungen funktionieren!

Der Abgabetermin für alle ist:

08.11.2016, 8:00

Eingang in meiner Inbox.