

Übung 10

Thema: Interface / Dateiverarbeitung

Aufgabe 1:

Eine Queue oder auch Warteschlange genannt ist eine öfter in der Informatik vorkommende Datenstruktur. Es lassen sich Werte (int) zu einer Queue hinzufügen und wieder entfernen. Es werden immer die zuerst hinzugefügten Werte entfernt. Dieses Verwaltungsprinzip nennt man deswegen auch first-in-first-out (FIFO).

Warteschlangen werden zum Beispiel bei der Prozessverwaltung in Betriebssystemen verwendet. Der zuerst hinzugefügte - wartende - Prozess, bekommt als nächstes wieder Prozessorzeit zugeordnet, vor allen nach ihm hinzugefügten Prozessen.

Entwerfen Sie ein Interface Queue, mit denen Werte hinzugefügt und wieder weggenommen werden können. Ebenso soll eine Methode existieren, die überprüft, ob die Queue leer ist oder nicht.

Interface Queue

All Known Implementing Classes:

[ArrayQueue](#)

```
public interface Queue
```

Eine Queue oder Warteschlange ist eine Sammlung von Daten (hier int-Werte). Es kann ein neuer Wert hinzugefügt oder der zuerst hinzugefügte Wert wieder entnommen werden: first-in-first-out Prinzip (FIFO).

[Zurück zum Aufgabentext](#)

Author:

pape

Method Summary

void	add (int wert)	Fügt den gegebenen wert zu dieser Queue hinzu.
boolean	isEmpty ()	Gibt genau true zurück, wenn in dieser Queue mindestens ein Wert enthalten ist.
int	remove ()	Entfernt den zuerst zu dieser Queue hinzugefügten Wert und gibt ihn zurück.

Implementieren Sie eine Klasse ArrayQueue. Sie besitzt Array als eine Instanzvariable.

Wenn mehr Werte in das Array hinzugefügt werden soll, als es fassen kann, dann soll das Array dieser Queue-Implementierung automatisch vergrößert werden: es kann zum Beispiel durch ein doppelt so großes Feld ersetzt werden.

Testen Sie Ihre Implementierung ausgiebig. Insbesondere anfangs mit einem kleinen Feld und mehreren Add- und Remove-Operationen.

Class ArrayQueue

```
java.lang.Object
└─ de.hska.java.aufgaben.interfaces.ArrayQueue
```

All Implemented Interfaces:

[Queue](#)

```
public class ArrayQueue
    extends java.lang.Object
    implements Queue
```

Implementiert eine Queue mit einem Feld. Wenn das Feld zu klein für die hinzugefügten Werte sind, dann wird dessen Grösse verdoppelt.

[Zurück zum Aufgabentext](#)

Author:

pape

Constructor Summary

[ArrayQueue](#)()

Erzeugt eine ArrayQueue mit initial 100 möglichen Werten.

[ArrayQueue](#)(int groesse)

Erzeugt eine ArrayQueue mit initial groesse möglichen Werten.

Method Summary

void [add](#)(int wert)

Fügt den gegebenen wert zu dieser Queue hinzu.

boolean [isEmpty](#)()

Gibt genau true zurück, wenn in dieser Queue mindestens ein Wert enthalten ist.

int [remove](#)()

Entfernt den zuerst zu dieser Queue hinzugefügten Wert und gibt ihn zurück.

Aufgabe 2:

Speichern Sie die Daten der Queue in eine Datei, wenn die Anwendung verlassen wird. Sorgen Sie dafür, dass die Daten bei einem erneuten Aufruf der Anwendung wieder zur Verfügung stehen.