

Deque de taille fixe

La classe `DequeConstraint` permet de représenter des deque de taille fixe. Lorsqu'un élément est ajouté au deque il est inséré au début, les éléments suivants sont décalés d'une position et le dernier élément sort du deque. Par exemple, soit une instance de cette classe pouvant contenir 3 éléments dans laquelle les éléments 1, 2 et 3 ont été insérés:

- Situation initiale: [3 2 1]
 - Insertion de la valeur 4: [4 3 2] (1 est éjecté)
1. Implanter la classe `DequeConstraint` en définissant (au moins):
 - Un attribut `private Object tableau[]` (contenant les éléments du deque),
 - Un constructeur `public DequeConstraint(int taille)`
 - Une méthode `public Object ajouter(Object element)` permettant d'insérer un élément dans le deque et retournant l'élément ainsi éjecté.

Rem: le décalage des éléments peut être optimisé au moyen d'un modulo, mais ce n'est pas indispensable.
 2. L'interface `Iterable` permet d'utiliser la boucle `for` simplifiée de Java 1.5 sur les objets dont les classes implantent cette interface.
 - L'interface `Iterable` déclare une méthode `Iterator iterator()`.
 - L'interface `Iterator` déclare les méthodes `boolean hasNext()`, `Object next()` et `void remove()` (l'implantation de `remove()` est optionnelle).

Implanter l'interface `Iterable` dans la classe `DequeConstraint`. Ne pas implanter l'opération `remove()`.
 3. Soit d un `DequeConstraint`, afficher ses éléments.

Deque de taille fixe

La classe `DequeConstraint` permet de représenter des deque de taille fixe. Lorsqu'un élément est ajouté au deque il est inséré au début, les éléments suivants sont décalés d'une position et le dernier élément sort du deque. Par exemple, soit une instance de cette classe pouvant contenir 3 éléments dans laquelle les éléments 1, 2 et 3 ont été insérés:

- Situation initiale: [3 2 1]
 - Insertion de la valeur 4: [4 3 2] (1 est éjecté)
1. Implanter la classe `DequeConstraint` en définissant (au moins):
 - Un attribut `private Object tableau[]` (contenant les éléments du deque),
 - Un constructeur `public DequeConstraint(int taille)`
 - Une méthode `public Object ajouter(Object element)` permettant d'insérer un élément dans le deque et retournant l'élément ainsi éjecté.

Rem: le décalage des éléments peut être optimisé au moyen d'un modulo, mais ce n'est pas indispensable.
 2. L'interface `Iterable` permet d'utiliser la boucle `for` simplifiée de Java 1.5 sur les objets dont les classes implantent cette interface.
 - L'interface `Iterable` déclare une méthode `Iterator iterator()`.
 - L'interface `Iterator` déclare les méthodes `boolean hasNext()`, `Object next()` et `void remove()` (l'implantation de `remove()` est optionnelle).

Implanter l'interface `Iterable` dans la classe `DequeConstraint`. Ne pas implanter l'opération `remove()`.
 3. Soit d un `DequeConstraint`, afficher ses éléments.