DEQUE

DESCRIPTION :

Un « deque » est une suite d’objets de même type, possédant un ordre bien précis, et dont le nombre est variable

(Un « deque » peut être vide !)

L’ajout se fait en tête ou en queue

Le retrait se fait en tête ou en queue

INTERFACE (AVEC JAVADOC):

**import** java.util.NoSuchElementException;

public interface Deque<E> {

/\*\*

\* renvoie le nombre d'elements qui se trouvent dans la decque

\* @return nombre d'elements

\*/

public **int size();**

/\*\*

\* verifie si la decque est vide

\* @return true si la decque est vide, false sinon

\*/

public **boolean isEmpty();**

/\*\*

\* insère l’élément au début de la decque

\* @throws NullPointerException si l’element est nul

\*/

public **void addFirst(E e) throws NullPointerException**;

/\*\*

\* ajoute un element en fin de la decque

\* @throws NullPointerException si l’element est nul

\*/

public **void addLast(E e) throws NullPointerException**;

/\*\*

\* *recupere et supprime le premier element de la deque*

\* @return la tete de la decque

\* @throws NoSuchElementException si la decque est vide

\*/

public **E removeFirst() throws NoSuchElementException**;

/\*\*

\* *recupere et supprime le dernier element de la deque*

\* @return la queue de la decque

\* @throws NoSuchElementException si la decque est vide

\*/

public **E removeLast() throws NoSuchElementException**;

/\*\*

\* *recupere la tete de la deque*

\* @return la tete de la decque

\* @throws NoSuchElementException si la decque est vide

\*/

public **E getFirst() throws NoSuchElementException**;

/\*\*

\* *recupere la queue de la deque*

\* @return la queue de la decque

\* @throws NoSuchElementException si la decque est vide

\*/

public **E getLast() throws NoSuchElementException**;

}

IMPLEMENTATION CHOISIE :

Table circulaire redimensionnable

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 1 | 2 | 3 |
| ‘a’ | ‘b’ |  |  |

tete

queue

JEU DE TESTS (sous forme d’un scénario avec au départ une deque vide) :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N°  test | Instruction | Exception  ? | return  attendu | deque.size() attendu | deque.toString()  attendu |
| 1 | deque.addFirst('a') |  |  | 1 | a |
| 2 | deque.addLast('b') |  |  | 2 | a b |
| 3 | deque.getFirst() |  | a | 2 | a b |
| 4 | Deque.getLast() |  |  | 2 | a b |
| 5 | deque.removeFirst() |  | a | 1 | b |
| 6 | deque.removeLast() |  | b | 0 |  |
| 7 | deque.removeFirst() | NoSuchElementException |  |  |  |
| 8 | deque.removeLast() | NoSuchElementException |  |  |  |
| 9 | deque.getFirst() | NoSuchElementException |  |  |  |
| 10 | deque.getLast() | NoSuchElementException |  |  |  |
| 11 | deque.addFirst('c') |  |  | 1 | c |

COUTS DES DIFFERENTES METHODES :

|  |  |
| --- | --- |
| size() | O(1) |
| isEmpty() | O(1) |
| addFirst() | O(1) |
| addLast() | O(1) |
| removeFirst() | O(1) |
| removeLast() | O(1) |