

Esercitazione n.2

Utilizzando le classi StackArray e QueueArray viste a lezione completare le classi **Esercizio1.java** ed **Esercizio2.java** specificate di seguito, rispettando il seguente vincolo:

per i metodi di entrambe le classi è possibile utilizzare solamente strutture dati locali di appoggio di tipo StackArray

Vi vengono fornite le classi TestEsercizio1.java e TestEsercizio2.java per testare i metodi delle due classi

NOTA: per poter compilare correttamente le classi StackArray.java QueueArray.java Esercizio1.java ed Esercizio2.java devono stare nella stessa directory

Consegnare entro domenica 2 Novembre 2008

Esercitazione n.2

Primo Esercizio: completare l'implementazione della classe

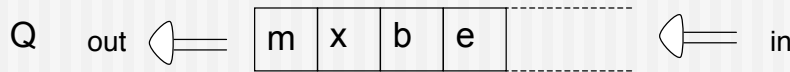
```
public class Esercizio1 {
    // pre: Q non nulla
    // post: ritorna una stringa che contiene tutti gli elementi in Q
    //       separati da uno spazio. L'ordine degli elementi nella stringa
    //       deve essere corrispondere all'ordinamento FIFO della coda.
    //       Il metodo non deve alterare lo stato originale di Q
    public static String toString(QueueArray Q) {...}

    // pre: Q non nulla
    // post: scambia l'ordine dei primi due elementi in coda, lasciando
    //       inalterati gli altri elementi
    // NOTA: se Q ha meno di due elementi allora rimane inalterata
    public static void scambia(QueueArray Q) {...}

    // pre: Q non nulla
    // post: ritorna uno stack che rappresenta fedelmente lo stato attuale di
    //       Q, lasciando inalterato lo stato di Q stessa
    // NOTA: rappresenta fedelmente significa che, dopo l'esecuzione del
    //       metodo, l'ordine di uscita degli elementi dallo stack e da Q
    //       deve essere lo stesso.
    public static StackArray toStack(QueueArray Q) {...}
}
```

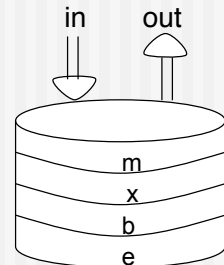
Esercitazione n.2

Primo Esercizio: esempio

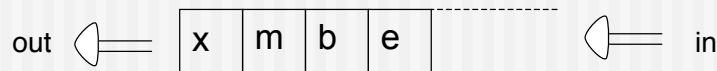


toString(Q) ritorna la stringa: "m x b e"

toStack(Q) ritorna uno stack il cui stato è il seguente:



scambia(Q) modifica la coda come segue:



Primo Esercizio: consegnare

- la classe `Esercizio1.java` completa
- la prova di correttezza del metodo `toString`

Esercitazione n.2

Secondo esercizio: completare l'implementazione della classe

```
public class Esercizio2 {
    // pre: S non nullo
    // post: ritorna una stringa che contiene tutti gli elementi in S
    //       separati da uno spazio. L'ordine degli elementi nella stringa
    //       deve essere corrispondere all'ordinamento LIFO dello stack.
    //       Il metodo non deve alterare lo stato originale di S
    public static String toString(StackArray S) {...}

    // pre: S non nullo
    // post: scambia l'ordine dei primi due elementi nello stack, lasciando
    //       inalterati gli altri elementi. Se S ha meno di due elementi allora
    //       rimane inalterato
    public static void scambia(StackArray S) {...}

    // pre: S non nullo
    // post: ritorna una coda che rappresenta fedelmente lo stato attuale di
    //       S, lasciando inalterato lo stato di S stesso
    // NOTA: rappresenta fedelmente significa che, dopo l'esecuzione del
    //       metodo, l'ordine di uscita degli elementi dalla coda e da S
    //       deve essere lo stesso.
    // OSS: per questo metodo e' possibile utilizzare una coda di supporto
    public static QueueArray toQueue(StackArray S) {...}
}
```

Esercitazione n.2

Secondo Esercizio: esempio

toString(S) ritorna la stringa: "v d t k"

toQueue(S) ritorna una coda il cui stato è il seguente:



scambia(S) modifica lo stack come segue:

Secondo Esercizio: consegnare

- la classe `Esercizio2.java` completa
- la prova di correttezza del metodo `toQueue`

