

**UNIVERSITÀ DEGLI STUDI DELL'INSUBRIA**  
**Dipartimento di Scienze Teoriche e Applicate**  
**Corso di Laurea in Informatica**  
**Insegnamento di Progettazione del Software**  
**Prof. Sandro Morasca**

**ESERCIZIO**

Si scriva una classe Java che gestisce una pila bifronte di numeri interi, ovvero una pila con due estremità disponibili per l'aggiunta e l'estrazione di elementi interi. Ad entrambe le estremità la politica utilizzata è quella impiegata nelle pile (ultimo aggiunto-primo estratto). In particolare, deve essere possibile effettuare le seguenti operazioni

- PilaBifronte: creare una pila bifronte vuota;
- Set • addSx: aggiungere un elemento all'estremità *sinistra* della pila bifronte;
- Set • delSx: rimuovere un elemento dall'estremità *sinistra* della pila bifronte
  - se la pila non è vuota: se la pila è vuota si deve restituire un'eccezione;
- frontSx: restituire l'elemento dall'estremità *sinistra* della pila bifronte
  - se la pila non è vuota: se la pila è vuota si deve restituire un'eccezione;
- addDx: aggiungere un elemento all'estremità *destra* della pila bifronte;
- delDx: rimuovere un elemento dall'estremità *destra* della pila bifronte
  - se la pila non è vuota: se la pila è vuota si deve restituire un'eccezione;
- frontDx: restituire l'elemento dall'estremità *destra* della pila bifronte
  - se la pila non è vuota: se la pila è vuota si deve restituire un'eccezione;
- eVuota: verificare se la pila bifronte è vuota;
- ePiena: verificare se la pila bifronte è piena;
- numeroElementi: contare il numero di elementi della pila bifronte.

Ad esempio, data la pila bifronte

22	7	12	25	4
----	---	----	----	---

l'aggiunta dell'elemento 13 all'estremità sinistra produrrà la pila bifronte

13	22	7	12	25	4
----	----	---	----	----	---

e poi la rimozione di un elemento dall'estremità destra produrrà la seguente pila bifronte

13	22	7	12	25
----	----	---	----	----

**Attenzione.** Se si sceglie una soluzione che fa uso di un array per immagazzinare gli elementi della pila bifronte, le funzioni addSx e addDx devono stampare un messaggio nel caso in cui vengano chiamate quando l'array è già completamente riempito. Si vuole tuttavia richiamare l'attenzione sul fatto che, in un'implementazione che utilizzi gli array, la seguente pila bifronte non è completamente piena.

87	45	24															
----	----	----	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Pertanto, l'aggiunta dell'elemento 31 all'estremità sinistra produrrà la pila bifronte

31	87	45	24														
----	----	----	----	--	--	--	--	--	--	--	--	--	--	--	--	--	--



```
public class PilaBifronte {
```

```
...
```

```
public void addSx(int x)
```

```
...
```

```
public void delSx() throws  
    EccezionePila {...}
```

```
public int frontSx() throws  
    EccezionePila {...}
```

```
...
```

```
public boolean eVuota() {...}
```

```
public boolean ePiena() {...}
```

```
public int numeroElementi() {...}
```