

# Analisi e progettazione del software

## Compito di metà corso

### 25 novembre 2014

Si intende realizzare una classe, chiamata `MultiPila`, che gestisca al suo interno  $n$  pile di interi indipendenti. Sulla classe `MultiPila` si eseguono le stesse 4 operazioni della classe `pila` (cioè `Push()`, `Pop()`, `Top()` e `IsEmpty()`), con la differenza che ogni operazione avrà un parametro in più (un intero con valore da 0 a  $n - 1$ ) che determina a quale delle  $n$  pile l'operazione si applica.

Inoltre, la classe dovrà avere un costruttore che riceve un parametro intero che rappresenta il numero  $n$  di pile, con valore di *default* pari da 1.

Si richiede di sviluppare la classe `MultiPila` utilizzando un unico vettore dinamico (senza usare `vector`), di dimensione  $dim \cdot n$ , con  $dim$  inizialmente pari a 100, nel modo seguente. La pila 0 occuperà le locazioni da 0 a  $dim - 1$ , la pila 1 quelle da  $dim$  a  $2 \cdot dim - 1$ , e così via fino alla pila  $n - 1$  che occuperà le locazioni da  $(n - 1) \cdot dim$  a  $n \cdot dim - 1$ .

Quando una qualsiasi pila supera la dimensione  $dim$ , il vettore viene riallocato di dimensione doppia e lo stato di tutte le pile salvato nel nuovo vettore.

**Esercizio 1 (punti 5)** Scrivere la definizione della classe `MultiPila`, prevedendo anche il selettore del numero di pile  $n$ . Si gestiscano tutte le precondizioni tramite la funzione `assert()` (per tutti gli esercizi).

**Esercizio 2 (punti 8)** Scrivere la definizione del costruttore e dei metodi `Push()`, `Pop()`, `Top()` e `IsEmpty()`.

**Esercizio 3 (punti 6)** Scrivere la definizione del costruttore di copia e del distruttore. Si tralasci invece la definizione dell'operatore di assegnazione.

**Esercizio 4 (punti 5)** Scrivere l'operatore di output (*friend*) della classe che scrive l'oggetto racchiudendo tra parentesi quadre la lista delle singole pile, a loro volta scritte come lista di elementi (con l'elemento affiorante a destra), racchiuse tra parentesi tonde.

Come esempio, si consideri l'oggetto composto dalle seguente 4 pile:

4			3
-3	-11		-6
-5	0		5
0	32		11

esso dovrà essere scritto: `[(0,-5,-3,4),(32,0,-11),(),(11,5,-6,3)]`

**Esercizio 5 (punti 6)** Scrivere una funzione esterna (non *friend*) che riceva come parametri un oggetto di tipo `MultiPila` ed un intero  $k$  (precondizione:  $0 \leq k < n$ ) e trasferisca nella pila  $k$  tutti gli elementi presenti nelle altre pile dell'oggetto, in ordine qualunque.

Ad esempio, se l'oggetto passato è quello riportato sopra e  $k = 3$ , l'oggetto risultante deve essere come il seguente: `[( ),( ),( ),(11,5,-6,3,4,-3,-5,0,-11,0,32)]`