

LISTE, PILE, CODE

Angelo Di Iorio

Università di Bologna

Esercizio 1

- Implementare una classe Java per gestire un lista di interi, su cui sono possibili le seguenti operazioni:
 - Inizializzazione lista vuota
 - Inserimento di un intero nella posizione i -esima
 - se $i==1$ -> in testa alla lista; se $i==2$ in seconda posizione, etc.
 - ritorna TRUE se l'inserimento è andato a buon fine, FALSE altrimenti
 - Cancellazione dell'intero nella posizione i -esima
 - se $i==1$ cancella l'intero in testa alla lista, se $i==2$ il secondo, etc.
 - ritorna TRUE se la cancellazione è andata a buon fine, FALSE altrimenti
- Realizzazione basata su una *lista monodirezionale con sentinella*

Esercizio 2

- Data una lista L di interi si vuole togliere da L ogni elemento pari e inserirlo in una nuova lista M, mantenendo in entrambe le liste l'ordine originario degli elementi.
- Usare (estendere) la classe `ArrayList` di Java

PILE, CODE

Angelo Di Iorio

Università di Bologna

Esercizio 1

- Implementare una classe Java per gestire una pila di oggetti generici
- Realizzare la pila tramite puntatori
- Implementare anche il metodo `toString()` che restituisce il contenuto della pila come stringa (invocando il metodo `toString()` degli oggetti contenuti in essa)

Esercizio 1

STACK

*% Restituisce **true** se la pila è vuota*

boolean isEmpty()

*% Inserisce *v* in cima alla pila*

push(ITEM *v*)

% Estrae l'elemento in cima alla pila e lo restituisce al chiamante

ITEM **pop**()

% Legge l'elemento in cima alla pila

ITEM **top**()

Esercizio 2

- Implementare una classe Java per gestire una coda di oggetti generici
- Realizzare la coda tramite un vettore circolare
- Implementare anche il metodo `print()` che restituisce il contenuto della pila come stringa (invocando il metodo `toString()` degli oggetti contenuti in essa)

Esercizio 2

QUEUE

*% Restituisce **true** se la coda è vuota*

boolean isEmpty()

*% Inserisce *v* in fondo alla coda*

enqueue(**ITEM** *v*)

% Estrae l'elemento in testa alla coda e lo restituisce al chiamante

ITEM dequeue()

% Legge l'elemento in testa alla coda

ITEM top()

Esercizio 3 (casa)

- Implementare una classe Java per gestire una coda di oggetti generici
- Realizzare la coda tramite puntatori