

Esame di Programmazione II

Appello di giorno 16 Luglio 2014
Università degli Studi di Catania - Corso di Laurea in Informatica

Testo della Prova

La corretta implementazione di ciascuno dei seguenti esercizi permette l'acquisizione di 7 punti. La corretta implementazione della classe come template è facoltativa e permette l'acquisizione di ulteriori 3 punti:

1. Si fornisca una classe C++, denominata `MyList<H>`, che implementi la seguente interfaccia `List<H>`, che rappresenta una lista linkata e contenente i seguenti metodi virtuali.
 - (a) `List<H>* ins(H *x)` aggiunge un nuovo elemento in testa alla lista e restituisce un puntatore ad un oggetto di tipo `List<H>`;
 - (b) `List<H>* del(H *x)` cancella l'elemento x dalla lista, se presente, e restituisce un puntatore ad un oggetto di tipo `List<H>`;
 - (c) `int search(H *x)` restituisce 1 se x è presente nella lista, 0 altrimenti;
 - (d) `void print()` è una procedura che stampa in output gli elementi della lista;Si crei quindi un'istanza di `MyList<int>` e si inseriscano al suo interno i seguenti valori:

23 4 6 8 12 21 5 9 7 3 16 2 24

Si esegua poi la cancellazione dei valori 4, 5, 16 e 2.

Infine si esegua la stampa dei valori inseriti nella lista.

L'output del programma sarà quindi:

23 6 8 12 21 9 7 3 24

```
template <class H> class List {
public:
    virtual List<H>* ins(H *x) = 0;
    virtual List<H>* del(H *x) = 0;
    virtual int search(H *x) = 0;
    virtual void print() = 0;
}
```

• • •

2. Si fornisca una classe derivata da `MyList<H>`, denominata `OrderedList<H>`, che rappresenti una lista ordinata di elementi. Si crei quindi un'istanza di `OrderedList<int>` e si eseguano le stesse operazioni eseguite nel precedente punto. L'output del programma sarà quindi:

3 6 7 8 9 12 21 23 24

• • •

3. Si fornisca una classe derivata da `MyList<H>`, denominata `MyStack<H>`, che rappresenti una struttura dati di tipo LIFO e che implementi la seguente interfaccia `Stack<H>`.

Si crei quindi un'istanza di `MyStack<int>` e si esegua la `push` dei seguenti valori:

23 4 6 8 12 21 5 9 7 3 16 2 24

Si eseguano poi 5 operazioni `pop()`

Infine si esegua la stampa dei valori inseriti nella lista.

L'output del programma sarà quindi:

9 5 21 12 8 6 4 23

```
template <class H> class Stack {
public:
    virtual void push(H *x) = 0;
    virtual H pop() = 0;
}
```

• • •

4. Si fornisca una classe derivata da `MyList<H>`, denominata `MyQueue<H>`, che rappresenti una struttura dati di tipo FIFO e che implementi la seguente interfaccia `Queue<H>`.

Si crei quindi un'istanza di `MyQueue<int>` e si esegua la `enqueue` dei seguenti valori:

23 4 6 8 12 21 5 9 7 3 16 2 24

Si eseguano poi 5 operazioni `dequeue()`

Infine si esegua la stampa dei valori inseriti nella lista.

L'output del programma sarà quindi:

21 5 9 7 3 16 2 24

```
template <class H> class Queue {
public:
    virtual void enqueue(H *x) = 0;
    virtual H dequeue() = 0;
}
```

• • •