Laboratorio di programmazione

Lezione X

Tatiana Zolo

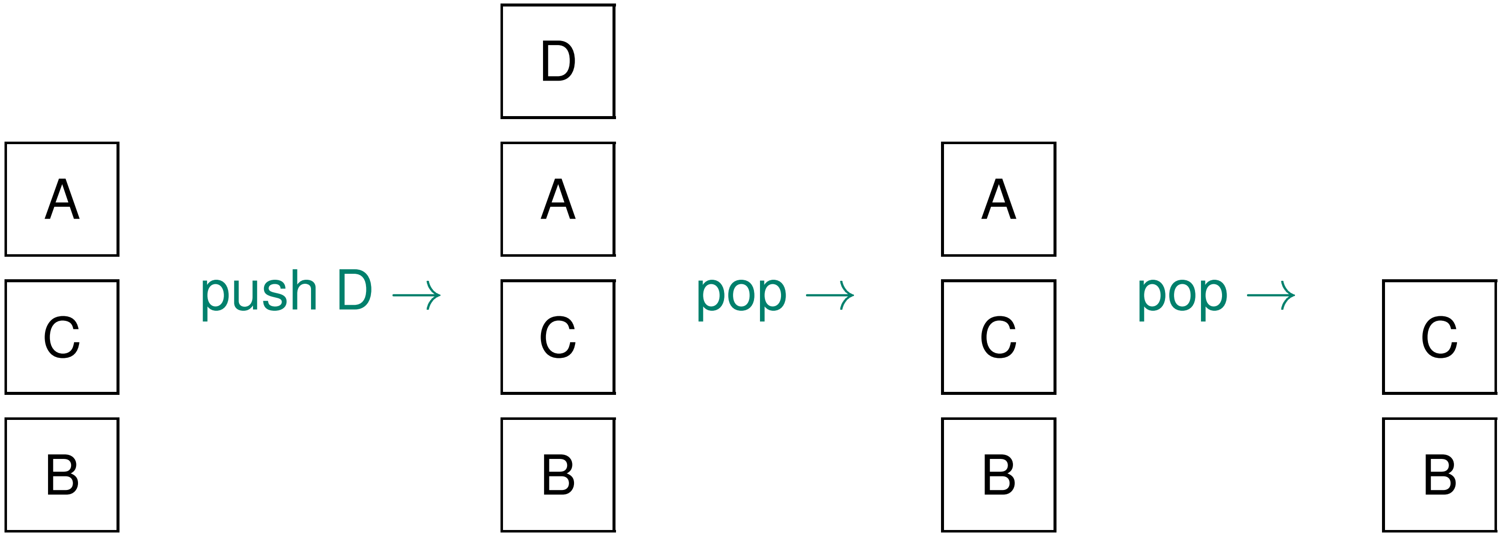
zolo@cs.unipr.it

Strutture dati: la pila

La pila (o stack) è un’astrazione che permette la memorizzazione ed il recupero di valori secondo una politica LIFO (*last-in-first-out*).

Le due operazioni fondamentali in una pila sono:

- push: inserisce un nuovo valore in testa;
- pop: elimina l’ultimo valore immesso.



ESERCIZI

1. Implementare la seguente classe “Pila” usando l’allocazione dinamica della memoria (new e delete). Fissare un numero max di elementi.

   ```
   class Pila {
   public:
     Pila();
     ~Pila();
     void push(int x);
     int pop();
     bool empty() const;
     unsigned int capacita() const;
     unsigned int num_elementi() const;
   private:
     int* A;
     int top; };
   ```

   Usare le eccezioni. (`pila.cc`)

ESERCIZI

2. Riscrivere la classe “Pila” aggiungendo la possibilità di raddoppiare la capacità della pila qualora venga fatta un’operazione di push su pila piena. Realizzare quanto chiesto con un metodo privato della classe, `raddoppia()`, invocato all’interno del metodo `push()`. (`pila_dinamica.cc`)

ESERCIZI

3. Scrivere un programma che legga una sequenza di espressioni aritmetiche da file e, utilizzando una pila di char, sia in grado di dire se le parentesi tonde di ogni espressione letta sono bilanciate e coerenti. Fornire come output la sequenza di espressioni seguite dal messaggio “Errata” qualora le parentesi non siano bilanciate e coerenti. Esempio:

   $((4 + 3) * 2) + 7$

   $(1 + (2 * 3)$

   Errata

   $(1 + (2 * 3))$

   $2 * (3 + 5) + 7 * (2 + 4))$

   Errata

   $2 * (3 + 5) + 7 * (2 + 4)$

   (`bilancia_parentesi.cc`)