

# Corso sul linguaggio Java

## Modulo JAVA7

### B2-Gestione di una pila

M. Malatesta B2-Gestione di una pila-15

1  
26/04/2014

## Prerequisiti

- Programmazione base in Java
- Utilizzo di classi e oggetti
- Tecnica di allocazione dinamica
- Concetto di pila e operazioni relative

M. Malatesta B2-Gestione di una pila-15

2  
26/04/2014

# Introduzione

In questa Unità vediamo come realizzare in pratica applicazioni che implementano una struttura astratta pila (stack), considerando per semplicità la pila formata da numeri interi.

Esaminiamo in particolare i seguenti casi:

- implementazione **sequenziale**
- implementazione **a lista concatenata**
- utilizzo di **Stack**

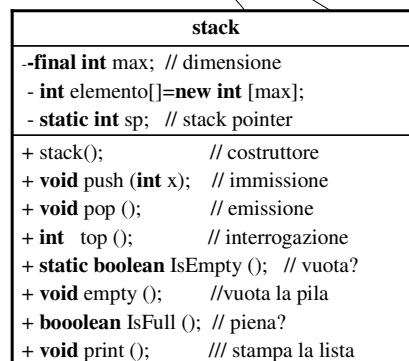
M. Malatesta B2-Gestione di una pila-15

3  
26/04/2014

## Implementazione sequenziale

**ATTIVITA'**: considerando le operazioni logiche presentate nello studio teorico, scrivere la UML della classe *stack* implementata con vettore.

La UML della classe è la seguente



M. Malatesta B2-Gestione di una pila-15

4  
26/04/2014

# Implementazione sequenziale

## - la classe *stack*

```
public class stack // implementazione classe stack
{
    final int max = 2;
    int elemento[] = new int [max];
    static int sp;
    public stack() { ... }
    public void push (int x) { ..... }
    public void pop() { ..... }
    public int top() { ..... }
    public static boolean IsEmpty() { ... }
    public void empty() { ..... }
    public boolean IsFull() { ..... }
    public void print() { ... }
    public static void main(String args[])
    { ..... }
} //end class
```

M. Malatesta B2-Gestione di una pila-15

5  
26/04/2014

# Implementazione sequenziale

## - il metodo *push()*

**ATTIVITA'**: implementare il metodo *push()* della classe *stack*.

```
public void push (int x)
{
    if (!IsFull())
    {
        elemento[sp]=x;
        sp++;
    }
    else
        System.out.println("Errore: pila piena");
}
```

M. Malatesta B2-Gestione di una pila-15

6  
26/04/2014

# Implementazione sequenziale

## - il metodo `pop()`

**ATTIVITA'**: implementare il metodo `pop()` della classe `stack`.

```
public void pop()
{
    if (!IsEmpty())
        sp--;
    else System.out.println("Errore: pila vuota");
}
```

**ATTIVITA'**: si lascia al lettore volenteroso l'implementazione dei restanti metodi!

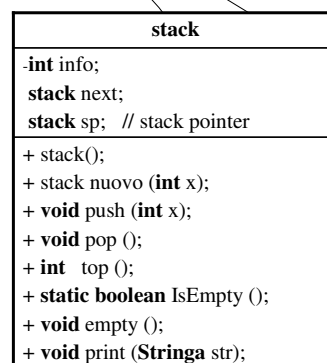
M. Malatesta B2-Gestione di una pila-15

7  
26/04/2014

# Implementazione a lista

**ATTIVITA'**: considerando le operazioni logiche presentate nello studio teorico, scrivere la UML della classe `stack` implementata a lista.

La UML della classe è la seguente



M. Malatesta B2-Gestione di una pila-15

8  
26/04/2014

# Implementazione a lista

## - il metodo `push()`

**ATTIVITA'**: implementare il metodo `push()` della classe `stack`.

```
public void push (int x)
{
    stack t = new stack();
    t.info = x;
    t.next = sp;
    sp = t;
}
```

M. Malatesta B2-Gestione di una pila-15

9  
26/04/2014

# Implementazione a lista

## - il metodo `pop()`

**ATTIVITA'**: implementare il metodo `pop()` della classe `stack`.

```
public void pop()
{
    if (!IsEmpty())
        sp = sp.next;
}
```

M. Malatesta B2-Gestione di una pila-15

10  
26/04/2014

# La classe **Stack** di Java

In Java è prevista la classe **Stack** che rappresenta una pila di oggetti i cui metodi sono illustrati di seguito.

METODO	EFFETTO
<b>Stack</b> <i>s</i> = new <b>Stack</b> ();	Costruttore
<b>boolean</b> empty()	Testa se lo stack è vuoto
<b>Object</b> peek()	Restituisce l'oggetto in cima senza estrarlo
<b>Object</b> pop()	Restituisce l'oggetto in cima e lo estrae
<b>Object</b> push( <b>Object</b> <i>elemento</i> )	Inserisce l'oggetto <i>elemento</i> in cima allo stack
<b>int</b> search( <b>Object</b> <i>o</i> )	Restituisce la posizione di <i>o</i> nello stack
<b>Object</b> pop ()	Dà il primo elemento della lista e lo rimuove
<b>int</b> size()	Restituisce il numero di elementi presenti

M. Malatesta B2-Gestione di una pila-15

11  
26/04/2014

## Argomenti

- Implementazione sequenziale
  - la classe *stack*
  - il metodo *push()*
  - il metodo *pop()*
- Implementazione a lista
  - il metodo *push()*
  - il metodo *pop()*
- La classe **Stack** di Java

M. Malatesta B2-Gestione di una pila-15

12  
26/04/2014