

INGEGNERIA DEL SOFTWARE

ESERCITAZIONE - INTRODUZIONE

09 OTTOBRE 2023

DAVIDE YI XIAN HU

EMAIL: DAVIDEYI.HU@POLIMI.IT

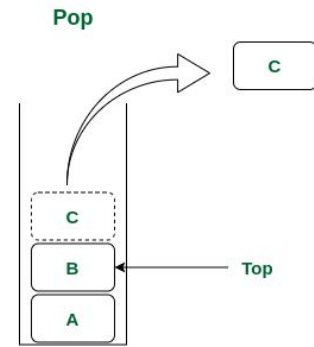
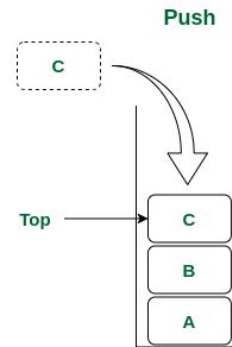


POLITECNICO
MILANO 1863

Esercizio 1 / Stack

Definire una classe stack di interi
(Struttura dati LIFO)

- 👉 La struttura dati ha una dimensione finita e arbitraria. Il valore di default e' 10.
- 👉 Metodo **pop** ritorna l'ultimo elemento e lo rimuove dallo stack.
- 👉 Metodo **push** aggiunge un elemento allo stack.



Esercizio 2 / String Literals

Illustrare l'effetto delle istruzioni in **rosso** sullo Heap.

Che uguaglianza c'è tra le quattro String? Utilizzare sia “==” che “s1.equals(s2)”

```
class StringLiterals {  
    public static void main(String args[]) {  
        String s1 = "abc";  
        String s2 = "abc";  
        String s3 = new String("abc");  
        String s4 = s3;  
    }  
}
```

Esercizio 3 / Parameter Evaluation

Illustrare e motivare il valore delle variabili `i`, `counter` e `counter2`.

```
class Counter {  
    int counter = 0;  
    public void increment(){  
        counter++;  
    }  
    public void incrementAndSet(int i){  
        i++;  
        counter=i;  
    }  
    public void incrementAndSet(Counter c){  
        c.counter++;  
        counter = c.counter  
    }  
}
```

```
class ParameterEvaluation {  
    public static void main(String args[]) {  
        Counter counter = new Counter();  
        counter.increment();  
        int i = 3;  
        counter.incrementAndSet(i);  
        Counter counter2 = new Counter();  
        counter2.incrementAndSet(i);  
        counter.incrementAndSet(counter2);  
    }  
}
```



Esercizio 4 / String Immutability

Cosa stampa questo programma?

```
public class StringDemo {  
    public static void main(String[] args){  
        String s1 = "Guess who";  
        String s2 = s1;  
        String s3 = "ABC";  
        s1 = s1 + " is back";  
        s1 = s3;  
        System.out.println(s1);  
        System.out.println(s2);  
    }  
}
```

Esercizio 5 / Persons and Students

Definire le classi **Person**, **Student** e **Grade**

- 👉 Una persona ha un nome, cognome e una data (`java.util.Date`).
- 👉 Uno studente è una persona con un id e una lista di voti.
- 👉 Un voto contiene punteggio e crediti.
- 👉 Lo studente espone due funzionalità:
 - Metodo ***meanGrade*** per calcolare media pesata.
 - Metodo ***canGraduate*** se è possibile che si laurei (crediti totali ≥ 180).



Esercizio 6 / Access Modifier

Completare il codice con gli opportuni modificatori di visibilità.

L'accesso alle variabili e ai metodi deve essere il più ristretto possibile ma non deve creare errori di compilazione.

public

visibile da qualsiasi parte del programma

private

visibile solo dall'interno della classe stessa

protected

visibile solo dalle classi dello stesso package e dalle sottoclassi

default

visibile dallo stesso package e dalle sottoclassi se sono nello stesso pacchetto

Esercizio 6 / Access Modifier

```
package a;
```

```
... class First {  
    ... int x;  
    ... int y;  
    ... void h() {  
        y = -1;  
    }  
}
```

```
... class Second extends First {  
    ... void f(int x) {  
        this.x = x;  
        h();  
    }  
}
```

```
package b;
```

```
imports a.*;
```

```
class Third {  
    public static void main(String[] s) {  
        Second z = new Second();  
        z.f(3);  
    }  
}
```

```
class Fourth extends First {  
    void g() {  
        h();  
    }  
}
```