

Fondamenti di Informatica – a.a. 2017-2018
Prova pratica di programmazione del 13 Settembre 2018

ORDINE

Per ordinare il contenuto di uno stack (creato come esemplare della classe `java.util.Stack` già esistente nelle API di Java 8) in modo che l'elemento più piccolo si trovi in testa allo stack, si vuole costruire la seguente classe

```
// OS.java -- Ordina Stack
import java.util.Stack;
public class OS
{ ... // eventuale parte privata
  public OS (Stack x) { ... }
  public void ordina () { ... }
}
```

sapendo che lo stack da ordinare contiene oggetti che realizzano l'interfaccia `java.lang.Comparable`.

- 1) Completare la classe `OS` scrivendo il codice del costruttore e del metodo `ordina()`. Per la soluzione del problema posto è consentito copiare il contenuto dello stack in un array, ordinare l'array e poi ricopiare l'array nello stack, ma verrà considerata soluzione ottima il procedere all'ordinamento utilizzando uno o più stack ausiliari e non array.
Indicare, in un commento all'inizio del codice del metodo `ordina()`, quale degli algoritmi noti è stato scelto per realizzare l'ordinamento (inserzione, selezione, mergesort, Bubblesort, ecc.).
- 2) Completare la classe `Main` che legga dall'ingresso standard un testo composto da stringhe, una per riga, lo inserisca in uno stack, ne ordini il contenuto utilizzando la classe `OS` e vuoti lo stack copiandone il contenuto sull'uscita standard.
Le righe del testo originale dovranno quindi comparire in ordine alfabetico crescente.

Al termine della prova il candidato dovrà lasciare nella directory di lavoro i seguenti file:

`OS.java` la classe per ordinare uno stack
`Main.java` l'applicazione per provare la classe `OS`.

I file `OS.java` e `Main.java` dovranno contenere come prima riga un commento con l'indicazione del **nome e cognome del candidato, numero di matricola, data della prova d'esame e numero della postazione di lavoro**.

Gli elaborati che non compilano non verranno valutati, durante la valutazione degli elaborati le classi sviluppate verranno provate con la sequenza di comandi:

```
$ rm *.class
$ javac Main.java
$ java Main <dati.txt >sorted.txt
$ sort dati.txt >ordinato.txt
$ diff sorted.txt ordinato.txt
# si verifica che i due file "risultato" e "ordinato"
# siano uguali riga per riga
```

FONDAMENTI DI INFORMATICA I - a.a. 2017-2018 - (prof. Sergio Canazza)

Nome e Cognome _____

Matricola _____ Postazione _____

☐ Consegno l'elaborato. ☐ Non consegno l'elaborato e mi ritiro dall'esame.

Firma _____