

Filtro — Analisi di un array di stringhe

Cognome: _____ Nome: _____ Matricola: _____ Calcolatore: _____

Specifiche: Nell'ambito della classe Java **Filtro**, definire i seguenti metodi di classe:

- un metodo **filtrati** che, ricevendo come parametri un array **v** (non nullo) di stringhe (non nulle) e un carattere **car**, calcola e restituisce il numero di elementi di **v** che contengono almeno una occorrenza del carattere **car**; l'array **v** non deve essere modificato da questa operazione;
- un metodo **filtro** che, ricevendo come parametri un array **v** (non nullo) di stringhe (non nulle) e un carattere **car**, crea e restituisce un array di stringhe che contiene tutti e soli gli elementi di **v** che contengono almeno una occorrenza del carattere **car**, nello stesso ordine in cui occorrono in **v**; l'array **v** non deve essere modificato da questa operazione.

Esempi: Si consideri l'array **v** = { "alfa", "beta", "gamma", "delta" }

- l'invocazione **Filtro.filtrati(v, 'e')** deve restituire l'intero **2** — infatti, gli elementi di **v** che contengono il carattere 'e' sono due, e precisamente "beta" e "delta"
- l'invocazione **Filtro.filtrati(v, 'a')** deve restituire **4**, e **Filtro.filtrati(v, 'z')** deve restituire **0**
- l'invocazione **Filtro.filtro(v, 'e')** deve restituire l'array (lungo **2**) { "beta", "delta" } — infatti, sia "beta" che "delta" contengono il carattere 'e', mentre non lo contengono né "alfa" né "gamma"
- l'invocazione **Filtro.filtro(v, 'a')** deve restituire l'array { "alfa", "beta", "gamma", "delta" }
- l'invocazione **Filtro.filtro(v, 'z')** deve restituire l'array { } — nessun elemento di **v** contiene 'z'

Che cosa deve essere fatto:

Sulla base delle specifiche dell'esercizio, definire una classe **Filtro** come segue:

- definire il metodo di classe pubblico **int filtrati(String[] v, char car)** — si assuma **v** non nullo, e ogni elemento di **v** non nullo;
- definire il metodo di classe pubblico **String[] filtro(String[] v, char car)** — si assuma **v** non nullo, e ogni elemento di **v** non nullo;
- definire una applicazione di test e, se necessario, altri metodi privati.

Che cosa è stato già fatto:

È possibile usare tutti i metodi della classe **String**; in particolare: il metodo d'istanza **int indexOf(char car)** (che calcola l'indice della prima occorrenza del carattere **car** nella stringa, e restituisce **-1** se la stringa non contiene nessuna occorrenza del carattere **car**).

Suggerimento:

Per la verifica, definire anche il seguente metodo per stampare un array di stringhe:

```
/* Stampa l'array di stringhe a, elemento per elemento. */
public static void stampa(String[] a) {
    // pre: a!=null
    int i;      // per la scansione di a
    for (i=0; i<a.length; i++)
        System.out.print(a[i] + " ");
    System.out.println();
}
```

Verifica:

La correttezza del codice scritto deve essere verificata mediante compilazione ed esecuzione, utilizzando degli opportuni insiemi di dati di ingresso.

Ciascuna degli insiemi di dati di ingresso usato per la verifica deve essere documentato da una invocazione di metodo e da un commento nel codice, che descrive l'insieme di dati di ingresso e motiva la scelta dell'insieme di dati di ingresso.