

Esercitazione 10 Ottobre 2019

Si considerino i tipi di dato astratti modificabili **ModString**, utilizzato per rappresentare stringhe mutabili, e **ModStringCollection**, utilizzato per rappresentare una collezione mutabile di oggetti di tipo **ModString**.

Il tipo di dati **ModString** ha, fra gli altri, i seguenti metodi

- **public int size()**, che restituisce la lunghezza della stringa;
- **public void update(char c, int num)**, che inserisce il carattere **c** nella posizione **num**, sovrascrivendo il carattere precedentemente in quella posizione;
- **public void remove(int num)**, che elimina il carattere in posizione **num**, eventualmente riposizionando in maniera adeguata i caratteri rimanenti.

Il tipo di dati **ModStringCollection** ha invece, fra gli altri, i seguenti metodi

- **public int occurrences(ModString str)**, che restituisce il numero di occorrenze della stringa **str** nella collezione;
- **public int size()**, che restituisce la cardinalità della collezione (il numero delle stringhe presenti);
- **public void insert(ModString str)**, che inserisce la stringa **str** nella collezione;
- **public int remove(ModString str, int num)**, che elimina dalla collezione al massimo **num** occorrenze della stringa **str** (e tutte le occorrenze se **num = 0**), e restituisce il numero degli elementi effettivamente rimossi;
- **public String[] getUnique()**, che restituisce gli elementi della collezione eliminando i duplicati.

Si completi la specifica dei due tipi di dato astratti, includendo una overview con la descrizione di un'istanza tipica e fornendo la specifica completa dei metodi, comprese le eventuali eccezioni lanciate.

Si definisca una implementazione dei tipi di dato astratto

ModString e **ModStringCollection**.

La prima deve utilizzare come struttura di supporto una rappresentazione del tipo

```
private final Character[] charEl;
```

mentre la seconda deve utilizzare come struttura di supporto una rappresentazione del tipo

```
private final Vector<ModString> str;  
private final Vector<Integer> occ;
```

In particolare, si definiscano la funzione di astrazione e l'invariante di rappresentazione.

Si dimostri la correttezza dell'implementazione proposta.