Laboratorio di algoritmi – Compito n. 0

Si crei una cartella **lists** e, al suo interno si definisca, nel modo illustrato a lezione, una classe **StringList** del package **lists**, con una classe interna **pubblica Node**.

La classe **Node** deve contenere i metodi:

- **next():** restituisce il valore del campo **next**;
- **addNext(String *elem*):** inserisce come successivo di **this** un nuovo nodo contenente ***elem***;
- **removeNext():** elimina il nodo successivo a **this**.

La classe **StringList** deve contenere i metodi:

- **firstNode();** restituisce il riferimento al primo nodo;
- **addHead(String *elem*)** aggiunge in testa un nuovo nodo contenente ***elem***;
- **removeFirst()** elimina il primo nodo;
- **ricerca(String *elem*)** restituisce il riferimento al primo nodo contenente ***elem***, oppure, se un tale nodo non esiste, il riferimento nullo;
- **write()** scrive sullo schermo la sequenza dei propri elementi;
- **write(String *fileName*)** scrive la sequenza dei propri elementi nel file di testo di nome ***fileName***, creandolo se non esiste, sovrascrivendolo se esiste: per la realizzazione si usi la classe predefinita **PrintWriter**;
- **read(String *fileName*)** **(metodo statico)** legge dal file di testo di nome ***fileName*** e costruisce e restituisce il corrispondente oggetto della classe **StringList** (ogni riga del testo è una stringa);
- **copy()** costruisce e restituisce una copia di **this**, con nodi tutti distinti dagli originali (quindi nuovi); **si realizzi copy in due versioni**:
  - **iterativa;**
  - **ricorsiva:** a tal fine potrebbe essere necessario definire la **copy()** ricorsiva nella classe Node, semplicemente richiamandola nel corrispondente metodo della classe **StringList**;

Si crei poi, all'esterno della cartella **lists**, una cartella **useLists**, e al suo interno si definisca una classe **UseStringList**, che utilizza la classe **StringList** per:

- costruire un oggetto della classe **StringList** leggendo i dati da un file di testo opportunamente preparato;
- scrive la lista sullo schermo;
- crea una copia di tale lista;
- verifica che modificando l'originale non si modifica la copia o viceversa;
- aggiunge dopo ogni stringa-elemento di lunghezza maggiore di 8 un nuovo elemento uguale a quello, ma con la lunghezza della stringa troncata a 8;
- toglie l'elemento immediatamente successivo ad ogni nodo contenente una stringa di lunghezza maggiore di 8;
- **elimina tutti gli elementi di lunghezza maggiore di 8 (attenzione: deve funzionare in tutti i casi possibili !)**;
- elimina tutti gli elementi della lista (che diventa così una lista vuota);

Nota: le azioni precedenti possono essere realizzate direttamente nel main, oppure in metodi statici separati richiamati dal main.

Si definisca infine, sempre nella cartella **useLists**, una classe **SortStringList** utilizzante **StringList** e contenente un metodo statico **selectionSort(StringList lis)** il quale, preso come argomento un oggetto della classe StringList, lo modifica ordinandone (lessicograficamente) la lista-contenuto:

- la realizzazione deve usare solo i metodi di **StringList** e non può creare una lista di appoggio;
- si devono solo scambiare i **contenuti** dei nodi, senza spostare i nodi stessi;
- il metodo di ordinamento deve essere quello dell'algoritmo **selectionSort** (**ordinam. per selezione**)