

Search and replace

Progetto d'esame per il corso di Programmazione a Oggetti

Prof. Giacomo Cabri

Traccia di progetto

REGOLE PER LO SVOLGIMENTO

Il progetto deve essere svolto usando il linguaggio Java e possedere i seguenti requisiti implementativi:

- Essere dotato di **interfaccia grafica** tramite cui interagire con il programma stesso.
- Sfruttare i meccanismi di **incapsulamento**, **ereditarietà** e **polimorfismo**.
- Per l'ereditarietà è possibile sfruttare **classi astratte** e **interfacce**; si considerano **escluse** le relazioni di ereditarietà diretta da classi di libreria Java.
- Sfruttare le classi di sistema Java per la gestione dell'**input/output**.
- Utilizzare i **generics**.
- Il programma deve essere eseguibile da **linea di comando**.

Il software deve essere accompagnato da pagine di **documentazione HTML** (ivi incluse le pagine generate tramite Javadoc) che descrivano le scelte di progetto effettuate e la struttura del sistema software.

Nel seguito del testo, i paragrafi evidenziati in **azzurro** sono **facoltativi**, e servono per differenziare il voto.

Lo svolgimento della parte **obbligatoria** contribuisce al voto per **20 punti**. Il contributo delle parti facoltative è riportato nelle rispettive descrizioni.

DESCRIZIONE DEL PROGETTO

La possibilità di fare sostituzioni testuali in modo automatico all'interno di un file o in più file contemporaneamente risulta molto utile in diverse situazioni, ad esempio quando si devono gestire diversi file di configurazione, o diversi file sorgenti, o anche semplici documenti. Una ulteriore utilità è quella di poter specificare un pattern invece di una stringa costante, cioè descrivere la struttura del testo che si vuole sostituire.

Il presente progetto si propone di descrivere e sviluppare una applicazione che abbia le seguenti funzionalità (dettagliate nei paragrafi successivi):

- Scelta dei file in cui operare le modifiche;
- Specifica delle modifiche;
- **Ricerca in sotto directory;**
- Visualizzazione delle modifiche effettuate in formato tabellare;
- Ricerca di informazioni nel risultato;
- **Stampa dei risultati.**

Scelta dei file in cui operare le modifiche

Come primo passo l'utente deve selezionare i file da modificare. Può essere un singolo file, oppure un insieme di file.

Opzionalmente, si deve dare la possibilità all'utente di selezionare tutti i file di una directory che hanno una data estensione [3 punti].

Specifiche delle modifiche

L'utente deve specificare due stringhe, una è quella che viene cercata, e l'altra è quella che va a sostituire la stringa trovata.

Si dia la possibilità di specificare un pattern, cioè uno schema della stringa da cercare. Ad esempio, il pattern può prevedere che il carattere '.' rappresenti qualsiasi carattere, quindi cercando la stringa "ca.a" si può trovare "casa", "cava", "cala", ecc. La definizione della sintassi dei pattern è lasciata allo studente.

Opzionalmente, si sfruttino le espressioni regolari per la ricerca di pattern all'interno del testo [5 punti].

Per implementare questa funzionalità si suggerisce di utilizzare il polimorfismo in Java.

Ricerca in sotto directory

Nel caso in cui il programma dia la possibilità all'utente di selezionare tutti i file di una directory che hanno una data estensione, deve permettere anche di eseguire le modifiche nei file che si trovano nelle sottodirectory di quella specificata [4 punti].

Per implementare questa funzionalità si suggerisce di utilizzare il polimorfismo in Java.

Visualizzazione delle modifiche effettuate in formato tabellare

Le modifiche apportate devono essere riportate all'utente in una tabella che elenca, una riga per ogni modifica effettuata:

- Nome del file
- Directory del file (se disponibile la ricerca in sottodirectory)
- Stringa trovata
- Stringa sostituita
- Posizione dall'inizio del file

Se la stringa cercata compare più volte nel file, si avranno più righe con lo stesso nome di file, la stessa stringa ma posizioni diverse.

Si dia la possibilità di **salvare** i risultati ottenuti e di aprire risultati salvati in precedenza. Nel caso in cui si tenti di salvare dei risultati in un file che esiste già, deve essere chiesto all'utente se desidera **sovrascrivere** il file esistente [2 punti].

Si offra all'utente, inoltre, la possibilità di **ordinare** le righe dei risultati in base ai valori di una colonna scelta [2 punti].

Ricerca di informazioni nei risultati

L'utente deve avere la possibilità di effettuare delle **ricerche** nei risultati correnti. La ricerca si basa su **testo libero** che può essere una parte del testo contenuto in una cella della tabella dei risultati.

La ricerca deve evidenziare la **prima cella** che contiene il testo cercato; l'applicazione deve permettere all'utente di **continuare** la ricerca per evidenziare man mano le celle **successive** che rispondono ai requisiti (ad esempio, tramite un bottone "successivo").

Stampa dei risultati

Si dia all'utente la possibilità di **stampare** i risultati correnti. Si sfruttino le classi di libreria Java per stampare tramite una delle stampanti configurate dal sistema operativo [2 punti].

MATERIALE UTILE

- Package `java.util.regex` per la gestione delle espressioni regolari. Sono disponibili diversi tutorial online sulle espressioni regolari.
- Interfaccia Java `java.awt.print.Printable` e classe Java `java.awt.print.PrinterJob` per stampare su stampante. Si possono trovare tutorial online.