

Esercizio

Scrivere un programma per la gestione di una videoteca.

Il programma deve essere in grado di tenere traccia dello stato corrente (presente/in prestito a x/smarrito da x/danneggiato da x a livello nn) e della storia delle transazioni a partire dalla sua acquisizione.

La chiave di ricerca e' singola: il titolo del video.

Il programma deve essere in grado di salvare uno stato su disco e di leggerlo all'avvio. Lo stato viene salvato su un file di testo denominato video.txt.

Il programma deve partire tentando di leggere uno stato esistente da file, nel caso esista elaborarlo in modo da ricostruire come corrente lo stato precedentemente salvato, porsi in attesa di comandi.

I comandi possibili (case sensitive) sono:

ESCI

Il comando provoca il salvataggio dello stato e la terminazione dell'esecuzione.

ELIMINA VIDEO

Il comando porta il programma a richiedere il titolo del video e rimuove il record opportuno. Nel caso il video non esista si genera un messaggio d'errore. L'esecuzione non termina ma ritorna al loop dei comandi.

TRANSAZIONE

Il comando porta il programma a richiedere in sequenza:

il nome del video;

il tipo di transazione;

la data nel formato descritto nella seconda pagina (formato file);

le informazioni aggiuntive nell'ordine definito e una per linea.

Una volta ottenute le informazioni necessarie si aggiungera' la transazione al record relativo al video.

N.B.1 L'inserimento del nome di un video non esistente per transazioni di tipo diverso dall'acquisizione deve generare un messaggio di errore e riportare al loop principale di comando senza inserire la nuova transazione.

N.B.2 L'inserimento del nome di un video gia' presente per una transazione di tipo acquisizione deve generare un messaggio di errore e riportare al loop principale di comando senza inserire la nuova transazione.

N.B.3 L'inserimento di una transazione logicamente non valida deve generare un messaggio di errore e riportare al loop principale di comando senza inserire la nuova transazione.

Le regole di non validita' delle transizioni sono le seguenti:

- 1) Non e' possibile restituire un video non in prestito.
- 2) Non e' possibile prestare un video in prestito/smarrito/danneggiato oltre il livello 50.
- 3) Non e' possibile inserire una transazione con data precedente a quella dell'ultima transazione inserita.

STORICO

Il comando porta il programma a richiedere il titolo del video e stampa a video la lista delle transazioni presenti nella storia del video. Nel caso il video non esista si genera un messaggio d'errore. L'esecuzione non termina ma ritorna al loop dei comandi.

LISTA

Il comando porta il programma a stampare la lista dei titoli dei video presenti nella videoteca¹.

STATO

Il comando porta il programma a richiedere il titolo del video e visualizza lo stato corrente (presente/in prestito a x/smarrito da x/danneggiato da x a livello nn).

NB4 Lo stato danneggiato da x a livello nn si visualizza solo nel caso in cui il danneggiamento sia superiore a 50 (cioe' renda il video indisponibile al prestito) e quindi il danneggiamento rappresenti l'ultima transazione presente.

Formato del file video.txt

Il file contiene i dati relativi a piu' video. Per ogni video il file contiene il seguente cluster di linee:

BEGIN

<titolo video>

<transazione 1>

...

<transazione n>

END

Il titolo e' una stringa che puo' contenere spazi.

Una transazione ha il seguente formato:

<data>

<tipo>

<informazioni accessorie dipendenti dal tipo della transazione>

Le date hanno il seguente formato:

gg/mm/aaaa

Le transazioni possono essere di tipo²:

acquisizione

prestito

restituzione

smarrimento

danneggiamento³

Una transazione di tipo acquisizione non ha informazioni aggiuntive.

Una transazione di tipo prestito, restituzione o smarrimento ha come informazioni aggiuntive:

Il nome del beneficiario/colpevole

Una transazione di tipo danneggiamento ha come informazioni aggiuntive:

Il nome del colpevole

Un numero da 1 a 100 indicante Il grado di danneggiamento

¹ Leggete con attenzione la documentazione delle classi Java che utilizzate.

² Le stringhe per il tipo della transazione sono identificative e case sensitive

³ Un danneggiamento implica comunque che il video e' stato restituito

Esercizio

Scrivere un programma per la gestione di una videoteca.

Il programma deve essere in grado di tenere traccia dello stato corrente (presente/in prestito a x/smarrito da x/danneggiato da x a livello yyyy) e della storia delle transazioni a partire dalla sua acquisizione.

La chiave di ricerca e' singola: il titolo del video.

Il programma deve essere in grado di salvare uno stato su disco e di leggerlo all'avvio. Lo stato viene salvato su un file di testo denominato video.txt.

Il programma deve partire tentando di leggere uno stato esistente da file, nel caso esista elaborarlo in modo da ricostruire come corrente lo stato precedentemente salvato, porsi in attesa di comandi.

I comandi possibili (case sensitive) sono:

ESCI

Il comando provoca il salvataggio dello stato e la terminazione dell'esecuzione.

ELIMINA VIDEO

Il comando porta il programma a richiedere il titolo del video e rimuove il record opportuno. Nel caso il video non esista si genera un messaggio d'errore. L'esecuzione non termina ma ritorna al loop dei comandi.

TRANSAZIONE

Il comando porta il programma a richiedere in sequenza:

il tipo di transazione;

il nome del video;

la data nel formato descritto nella seconda pagina (formato file);

le informazioni aggiuntive nell'ordine definito e una per linea.

Una volta ottenute le informazioni necessarie si aggiungera' la transazione al record relativo al video.

N.B.1 L'inserimento del nome di un video non esistente per transazioni di tipo diverso dall'acquisizione deve generare un messaggio di errore e riportare al loop principale di comando senza inserire la nuova transazione.

N.B.2 L'inserimento del nome di un video gia' presente per una transazione di tipo acquisizione deve generare un messaggio di errore e riportare al loop principale di comando senza inserire la nuova transazione.

N.B.3 L'inserimento di una transazione logicamente non valida deve generare un messaggio di errore e riportare al loop principale di comando **senza inserire la nuova transazione.**

Le regole di non validita' delle transizioni sono le seguenti:

- 4) Non e' possibile restituire un video non in prestito.
- 5) Non e' possibile prestare un video in prestito/smarrito/danneggiato oltre il livello medio.
- 6) Non e' possibile inserire una transazione con data precedente a quella dell'ultima transazione inserita.

STORICO

Il comando porta il programma a richiedere il titolo del video e stampa a video la lista delle transazioni presenti nella storia del video. Nel caso il video non esista si genera un messaggio d'errore. L'esecuzione non termina ma ritorna al loop dei comandi.

LISTA

Il comando porta il programma a stampare la lista dei titoli dei video presenti nella videoteca⁴.

STATO

Il comando porta il programma a richiedere il titolo del video e visualizza lo stato corrente (presente/in prestito a x/smarrito da x/danneggiato da x a livello nn).

NB4 Lo stato danneggiato da x a livello yyyy si visualizza solo nel caso in cui il danneggiamento sia superiore a medio (cioè renda il video indisponibile al prestito) e quindi il danneggiamento rappresenti l'ultima transazione presente.

Formato del file video.txt

Il file contiene i dati relativi a più video. Per ogni video il file contiene il seguente cluster di linee:

BEGIN

<titolo video>

<transazione 1>

...

<transazione n>

END

Il titolo è una stringa che può contenere spazi.

Una transazione ha il seguente formato:

<tipo>

<data>

<informazioni accessorie dipendenti dal tipo della transazione>

Le date hanno il seguente formato:

mm-gg-aa

Le transazioni possono essere di tipo⁵:

acquisizione

prestito

restituzione

smarrimento

danneggiamento⁶

Una transazione di tipo acquisizione non ha informazioni aggiuntive.

Una transazione di tipo prestito, restituzione o smarrimento ha come informazioni aggiuntive:

Il nome del beneficiario/colpevole

Una transazione di tipo danneggiamento ha come informazioni aggiuntive:

Il nome del colpevole

Una stringa (leggero/medio/pesante) indicante il grado di danneggiamento

⁴ Leggete con attenzione la documentazione delle classi Java che utilizzate.

⁵ Le stringhe per il tipo della transazione sono identificative e case sensitive

⁶ Un danneggiamento implica comunque che il video è stato restituito

Esercizio

Scrivere un programma per la lettura, l'ordinamento e il salvataggio di gruppi di record personali di tipo, dimensione e contenuto variabile.

Lo stato viene letto da un file di testo denominato record.txt e salvato su un file di testo denominato record_new.txt.

Il programma deve leggere tutti i record dal file formattato come descritto sotto e scriverli in modo che il nuovo file risulti formattato come descritto sotto.

N.B. Una data si intende costituita da una stringa di esattamente otto caratteri secondo il formato aaaammgg quindi si possono ordinare alfabeticamente.

Il programma deve essere progettato in modo da permettere di aggiungere facilmente nuove tipologie di record.

Formato del file record.txt

Il file contiene una sequenza di cluster anche annidati di record.

Ogni cluster di record contiene

1. La stringa "INIZIO"
2. Il nome della persona
3. Il CF della persona
4. Uno o piu' tra
 - a. un record
 - b. l'inizio di un nuovo cluster
5. la Stringa "FINE"

Un record e' composto da:

1. una linea contenente il tipo del record
2. una data di inizio
3. un numero di linee aggiuntive (anche zero) variabili a seconda del tipo di record

Si definiscono i record di tipo:

1. "incidente"
2. "malattia"
3. "riposo"

Un record "incidente" contiene le seguenti righe aggiuntive:

1. il nome di una localita'
2. il nome di un ospedale
3. una data di dimissione

Un record "malattia" contiene le seguenti righe aggiuntive:

1. un numero che da' la durata in giorni

Un record "riposo" non contiene altre righe aggiuntive

Formato del file record_new.txt

<gruppo di record della persona 1 ordinati per date crescenti>

.

.

<.gruppo di record della persona n ordinati per date crescenti >

Un gruppo di record e' cosi' formato:

"INIZIO"

Nome della persona

CF della persona

<record 1>

.

.

<record n>

“FINE”

I record hanno lo stesso formato che hanno nel file record.txt.

Esercizio

Scrivere un programma che gestisca gli investimenti di un investitore.

Il programma deve gestire uno stato persistente tramite files di testo formattati come specificato piu' avanti.

Il programma deve permettere le seguenti operazioni:

1. AGGIORNA il valore di tutti gli investimenti di un dato tipo;
2. VISUALIZZA un singolo investimento;
3. ESCI dal programma.

Le parole in maiuscolo costituiscono i comandi da accettare da tastiera.

Un investimento e' identificato da alcuni campi :

1. un codice alfanumerico univoco;
2. un valore corrente (in Euro);
3. un tipo alfanumerico.

Il tipo (case sensitive) puo essere:

1. Azioni;
2. CD;
3. Obbligazioni.

Il tipo azioni non richiede ulteriori informazioni;

Il tipo CD contiene anche una singola data di maturazione;

Il tipo obbligazioni contiene un numero e una serie di date di maturazione delle cedole (tante quante era il numero precedente).

Il comando AGGIORNA richiedera' il tipo di investimento, un valore percentuale positivo o negativo e aggiornera' il valore corrente in EURO di tutti gli investimenti di quel tipo aggiungendo o sottraendo il valore ottenuto dal calcolo della percentuale fornita;

Il comando VISUALIZZA richiedera' il codice identificativo dell'investimento e ne visualizzera' i dati;

Il comando ESCI salvera' lo stato corrente su file e terminera' l'esecuzione.

FORMATO DEL FILE DI STATO

File di testo, un campo per linea

codice

valore

tipo

[campi opzionali in base al tipo]

Esercizio

Scrivere un programma che gestisca le informazioni relative alle precipitazioni piovose verificatesi in un anno su una zona suddivisa tramite una griglia quadrata 100x100.

Il programma deve gestire i comandi descritti nel seguito del testo.

LEGGI

Il programma richiederà il nome del file, ne leggerà i dati e li inserirà nello stato corrente.

STORICO

Il comando richiederà le coordinate x e y (numeri interi) del sito di cui si vogliono visualizzare i dati e fornirà la lista completa di tutti i dati presenti ordinata per date (dalla più remota alla più recente).

MEDIA

Il comando richiederà le coordinate x e y (numeri interi) del sito di cui si vogliono visualizzare i dati e fornirà la media aritmetica delle precipitazioni e la lunghezza in giorni del periodo di tempo cui il dato si riferisce.

FORMATO DEL FILE DI INPUT

File di testo, un campo per linea. I dati non sono necessariamente ordinati cronologicamente.

coordinata X

coordinata Y

“Giorno” giorno dell’anno (da 1 a 365)

“mm di pioggia” valore (virgola mobile)

Esercizio

Scrivere un programma che gestisca le informazioni relative ai livelli di micropolveri presenti nei diversi giorni dell'anno nella città di Genova. La città sarà suddivisa tramite una griglia quadrata 10x100.

Il programma deve gestire i comandi descritti nel seguito del testo.

LEGGI

Il programma richiederà il nome del file, ne leggerà i dati e li inserirà nello stato corrente.

ELENCO

Il comando richiederà le coordinate x e y (numeri interi) della zona di cui si vogliono visualizzare i dati e fornirà la lista completa di tutti i dati presenti in memoria relativamente alla zona richiesta. Tale visualizzazione dovrà essere ordinata per data (dalla più remota alla più recente).

MEDIA

Il comando richiederà le coordinate x e y (numeri interi) della zona di cui si vogliono visualizzare i dati e fornirà la media aritmetica del livello di micropolveri presenti nell'aria e la lunghezza misurata in numero di giorni del periodo di tempo cui il dato visualizzato si riferisce.

FORMATO DEL FILE DI INPUT

File di testo, un campo per linea. I dati non sono necessariamente ordinati cronologicamente.

coordinata X

coordinata Y

“Giorno ” giorno dell'anno (da 1 a 365)

“Microgrammi per metro cubo ” valore (virgola mobile)

Esercizio

Implementare un programma in grado di tenere traccia dei libri presenti in una biblioteca.

Per ogni libro il programma deve tenere traccia di due elementi informativi:

- Titolo del libro;
- Autore del libro.

I due campi del record hanno le seguenti proprietà:

- il titolo di un libro e' una chiave univoca, non possono esistere, cioe', due libri di autori diversi con lo stesso titolo;
- ogni titolo e' ad autore singolo, non si puo' quindi verificare la situazione che un libro possa essere attribuito a piu' di un autore.

Il programma deve poter:

- 1) Stampare una lista di tutti i libri presenti nella biblioteca raggruppati per autore e, all'interno dei gruppi di titoli pertinenti allo stesso autore, ordinati alfabeticamente per titolo;
- 2) Effettuare ricerca con chiave uguale al titolo e stampare il risultato (un singolo record viste l'univocita' della chiave titolo);
- 3) Effettuare ricerca con chiave uguale all'autore e stampare la lista di tutti i titoli pertinenti a quell'autore ordinati alfabeticamente per titolo;

Non e' necessario che il programma sia in grado di gestire uno stato persistente da un'esecuzione all'altra ma deve essere in grado di leggere uno o piu' files di record formattati come descritto di seguito in modo da costruirsi uno stato corrente su cui eseguire le procedure precedentemente descritte. Il numero di record per file non e' noto a priori. I record all'interno del file non sono ordinati e possono appartenere ad autori diversi.

Formato dei record.

File di testo, un campo per linea, zero o piu' campi per file.

<autore>

<titolo>

Esercizio

Implementare un programma in grado di tenere traccia del livello di radiazione presente in un insieme di zone di spazio tridimensionale (coordinate x, y e z). Ogni zona ha la forma di un parallelepipedo, i campioni sono spazati uniformemente e ci si aspetta di avere un numero di campioni prossimo al numero di punti caratterizzati da valori interi delle coordinate.

Le dimensioni massime di ogni zona di spazio non sono note a priori e possono cambiare da zona a zona.

Per ogni campione il programma deve tenere traccia di due elementi informativi:

- Il livello di radiazione;
- La data e ora in cui il valore e' stato campionato.

Il programma deve poter eseguire le seguenti operazioni.

4) Leggere un file di inizializzazione contenente dei record cosi' strutturati:

- a. Identificatore della zona di spazio (stringa);
- b. x massima;
- c. y massima;
- d. z massima.

5) Leggere dei files di dati contenenti dei record cosi' strutturati:

- a. Identificatore della zona di spazio;
- b. coordinata x;
- c. coordinata y;
- d. coordinata z.
- e. valore radiazione;
- f. data e ora di rilevazione (gg/mm/aaaa,hh:mm)

6) Effettuare ricerca con chiave uguale all'identificatore della zona di spazio e stampare i dati relativi ad essa (Lunghezza, larghezza e altezza della zona sulla prima linea e poi le coppie valore, data di rilevazione "affettando" lo spazio per piani);

Non e' richiesto che il programma sia in grado di gestire uno stato persistente da un'esecuzione all'altra ma deve essere in grado di leggere uno o piu' files di record formattati come descritto sopra in modo da costruirsi uno stato corrente su cui eseguire le procedure richieste. Il numero di record per file non e' noto a priori. I record all'interno del file possono appartenere a zone di spazio diverse ma comunque presenti nel file di inizializzazione.

Non e' richiesto gestire dati inconsistenti (es. coordinate al di fuori dell'intervallo definito all'inizializzazione).