

Specifiche realizzazione Tasks

27/12/2020

Descrizione metodologia realizzazione generale:

I task richiesti dalla challenge sono stati realizzati con il linguaggio di programmazione python e ognuno è stato salvato su file .py differente. Il file csv è stato salvato in locale e importato in ogni singolo task. I task richiesti sono stati scritti con l'ambiente di sviluppo nativo di python IDLE e testati prima di essere inseriti nel repository. All'interno del repository è presente anche il file csv stesso.

Task 1:

Al fine di realizzare le specifiche richieste, il programma è stato implementato secondo questa logica:

- 1) E' stato importato il modulo **csv** per poter operare con il file csv. Il file csv è stato importato con il nome **fileDati**. La variabile **csvHandler** è stata istanziata per poter operare sul file csv (in lettura) e quindi collegata tramite opportuno comando a **fileDati**.
- 2) Il programma richiede all'utente l'inserimento del nome del docente. Tale valore viene salvato nella variabile **docenteInserito**. Il programma richiede di inserire il nome e/o il cognome dal docente da ricercare con le iniziali maiuscole
- 3) Viene effettuato un controllo sulla variabile **docenteInserito**: se questa variabile rimane vuota, il programma estrarrà ogni riga (esclusa la prima la quale è saltata attraverso l'istruzione next() per evitare di mostrare a video l'intestazione) dal file csv e la salverà nella lista **dati** al fine di mostrare tutti i corsi presenti. Se la variabile **docenteInserito** non è vuota è quindi presente un nome o un cognome e il programma estrarrà dal file csv solo le righe che contengono nella colonna **Instructors** (corrispondente all'elemento con indice [4] nella riga estratta dal csv) la stringa inserita in **docenteInserito**, tali righe verranno inserite nella lista **dati**.
- 4) I valori presenti in **dati** vengono quindi mostrati a schermo.

Task 2:

Al fine di realizzare le specifiche richieste, il task 1 è stato modificato inserendo le seguenti features:

- 1) E' possibile ricercare piu' di un docente. Il programma richiede in input da parte dell'utente il numero di docenti da ricercare, tale valore viene convertito in tipo **int** e salvato nella variabile **numeroDocenti**. Come mostrato nel messaggio a schermo, se viene inserito il valore **0** verranno estratti tutti i corsi (e le relative informazioni) presenti nel file csv (a meno di eventuali filtri descritti di seguito) e inseriti in maniera analoga al task 1 in **dati**. Inserendo un numero finito, il sistema richiederà all'utente l'inserimento dei docenti da ricercare. Tali stringhe verranno inserite nella lista **listaDocenti**.

- 2) Viene richiesto all'utente se filtrare i risultati per anno solare. La variabile **filtroAnnoSolare** viene utilizzata come una sorta di *flag* per indicare al programma se l'utente desidera filtrare o meno i risultati per anno solare. Il programma indica all'utente di rispondere alla richiesta mediante valori *Y* o *N*. **NB: E' importante inserire esclusivamente i valori richiesti poiché al momento non è stato implementato un sistema di validazione dell'input dell'utente.** Se l'utente richiede di filtrare i risultati per anno solare il programma chiederà all'utente l'inserimento dell'anno per poi salvarlo all'interno della variabile **annoSolare**. Se l'utente non desidera utilizzare questo filtro il programma passerà alla prossima richiesta.
- 3) In maniera analoga viene richiesto all'utente se filtrare i risultati per anno di corso. La logica è la stessa utilizzata nel punto precedente, utilizzando in questo caso le variabili **filtroAnnoCorso** e **annoCorso**.
- 4) Viene avviata la ricerca sia nel caso in cui non è stato inserito alcun docente che nel caso sono presenti dei docenti inseriti. In entrambi i casi vengono controllati i *flag* **filtroAnnoSolare** e **filtroAnnoCorso** in modo da estrarre le righe dal file csv in cui le colonne **Launch Date** e/o **Year** contengono i valori da noi richiesti, memorizzati nelle variabili **annoSolare** e **annoCorso**; anche in questo caso le righe estratte vengono salvate in **dati**.
 Nel caso in cui sono presenti uno o più docenti, la **listaDocenti** viene ciclata: per ogni docente presente in tale lista viene estratta dal file csv ogni riga che contiene nel campo **Instructors** il nome e/o cognome del docente a cui stiamo facendo attualmente riferimento; anche in questo caso verranno effettuati i controlli su eventuali filtri di ricerca e quindi estratte le righe secondo la medesima logica descritta precedentemente. Prima di passare al prossimo docente il **fileDati** viene resettato attraverso il metodo **.seek** in modo da ritornare all'inizio del documento. Anche in questo caso le righe estratte vengono salvate in **dati**. E' stato scelto di estrarre le righe secondo questa logica in modo tale che in **dati** le righe relative ad un determinato docente saranno poste in posizione adiacente l'una all'altra, "accorpando" così i risultati.
- 5) Per evitare di visualizzare più volte lo stesso corso si è operato in questo modo: per prima cosa è stata istanziata una lista **corsiTrovati**, tale lista servirà per tenere traccia dei corsi già trovati e già mostrati. Ogni volta che viene estratta una riga da **dati** il numero del corso relativo a tale riga viene convertito in stringa e salvato nella variabile **corsoAttuale**. Viene quindi controllato se **corsoAttuale** non è presente in **corsiTrovati**. Se tale corso non è presente allora significa che è la prima volta che viene estratto; la riga corrispondente viene quindi estratta da **dati** e visualizzata a schermo secondo opportuna formattazione e, infine, il **corsoAttuale** viene aggiunto alla lista **CorsiTrovati**. Se il corso è già presente nella lista **corsiTrovati** la riga relativa a tale corso non viene visualizzata.

Task 3:

Al fine di realizzare le specifiche richieste, il task 2 è stato modificato per permettere il salvataggio dei risultati della ricerca in un file xls.

- 1) E' stato importato il modulo **xlsxwriter** per poter operare con i file xls
- 2) La logica del programma è analoga a quella descritta nel task 2. La differenza risiede nel fatto che in fase di acquisizione input viene richiesto all'utente se desidera salvare i risultati in un file xls.
- 3) Vengono aggiunte le variabili **rigaXls** e **colonnaXls**. Tali variabili serviranno come indici per righe e colonne per il file xls.
- 4) Se l'utente decide di salvare i risultati nel file xls viene creato un file **risultati ricerca.xlsx**.

Il programma scriverà come prima riga e relative colonne l'header della tabella che verrà creata all'interno del file. Viene incrementato l'indice di riga (incrementando quindi la variabile ***rigaXls***) per passare alla riga successiva del foglio excel ed essere pronti a scrivere gli eventuali risultati.

- 5) I risultati vengono scritti sul foglio elettronico nel momento in cui vengono visualizzati a schermo, rispettando quindi la logica dello scrivere una sola volta un corso che compare più volte tra i risultati estratti dal file csv
- 6) A termine della ricerca il file xls viene chiuso e quindi salvato nella directory dove sono presenti i file .py