**FILE DI PROGETTAZIONE**

**TIPI DI DATO/ STRUTTURE DATI**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Tipologia** | **Descrizione** | **Tipi/Campi/Valori** |
| Data | Struct | Serve per memorizzare le date | Giorno: int; Mese: int; Anno: int |
| MENU | Struct | Struct usata per la gestione del menù interattivo | VoceMenu: char[50];  Funzione: void(\*)() |
| Preferenza | Struct | Struct usata per gestire l’entità preferenza | IDUtente: int; IDProdotto:int; Tipo:short |
| PreferenzeAssociate | Struct | Serve per associare all’index di un elemento (utente o prodotto) il numero delle preferenze associate | Index: int; Numero: int |
| Prodotto | Struct | Struct usata per gestire l’entità prodotto | ID: int; Nome: char[L\_ARRAY\_CHAR];  Categoria: Categoria;  Prezzo: float; NumeroPezzi:int; Ditta: char[L\_ARRAY\_CHAR]; DataInserimento:Data;  Descrizione: char[L\_DESCRIZIONE]; |
| ProdottoStima | Struct | Struct usata per calcolare quanto un prodotto potrebbe piacere all'utente interessato | IndexProdotto: int;  Stima: float |
| Utente | Struct | Struct usata per gestire l’entità utente | ID: int; Email: char[L\_ARRAY\_CHAR]; Nome: char[L\_ARRAY\_CHAR];  Cognome: char[L\_ARRAY\_CHAR];  DataNascita: Data; DataIscrizione: Data; Sesso:bool;  NumeroTelefono: char[L\_NUMERO\_TELEFONO] |
| UtenteSimilarita | Struct | Struct usata per il calcolo degli utenti simili | IndexUtente: int; Similarita: float; Preferenze: Preferenza[PREFERENZE\_UTENTE];  NumeroPreferenze: int |
| Categoria | Typedef | Abbreviazione usata per indicare il nome della categoria | Char[L\_NOME\_CATEGORIA] |
| arrCategorie | Variabile globale | Contiene l’elenco delle categorie | Categoria[MAX\_CATEGORIE] |
| arrPreferenze | Variabile globale | Contiene l’elenco delle preferenze | Preferenza[MAX\_PREFERENZA] |
| arrProdotti | Variabile globale | Contiene l’elenco dei prodotti | Prodotti[MAX\_PRODOTTI] |
| arrUtenti | Variabile globale | Contiene l’elenco degli utenti | Utente[MAX\_UTENTI] |
| flCategorie | Variabile globale | File ad accesso binario delle categorie | FILE\* |
| flPreferenze | Variabile globale | File ad accesso binario delle preferenze | FILE\* |
| flProdotti | Variabile globale | File ad accesso binario dei prodotti | FILE\* |
| flUtenti | Variabile globale | File ad accesso binario degli utenti | FILE\* |
| NCategorie | Variabile globale | Numero di categorie nel sistema | int |
| NPreferenze | Variabile globale | Numero di preferenze nel sistema | int |
| NProdotti | Variabile globale | Numero di prodotti nel sistema | int |
| NUtenti | Variabile globale | Numero di utenti nel sistema | int |
| SalvaModificheFilePreferenze | Variabile globale | Indica se l'elenco delle preferenze è stato modificato, in tal caso verrà aggiornato il file specifico | bool |
| LastIDProdotto | Variabile globale | Memorizza l'ID dell'ultimo prodotto in memoria, ossia il piu' grande | int |
| LastIDUtente | Variabile globale | Memorizza l'ID dell'ultimo utente in memoria, ossia il piu' grande | int |
| COLORE\_BIANCO | Costante | Corrisponde al colore bianco visualizzato in console | 7 |
| COLORE\_ROSSO | Costante | Corrisponde al colore rosso visualizzato in console | 4 |
| FRECCIA\_GIU | Costante | Indica il numero equivalente quando si preme il tasto ‘freccia giù’ | 72 |
| FRECCIA\_SU | Costante | Indica il numero equivalente quando si preme il tasto ‘freccia su’ | 80 |
| L\_ARRAY\_CHAR | Costante | Lunghezza di una generica stringa per memorizzare i dati | 30 |
| L\_DESCRIZIONE | Costante | Lunghezza della descrizione | 200 |
| L\_FILTRO\_RICERCA | Costante | Lunghezza del filtro di ricerca | 100 |
| L\_NOME\_CATEGORIA | Costante | Lunghezza del nome della categoria | 20 |
| L\_NUMERO\_TELEFONO | Costante | Lunghezza del numero di telefono | 10 |
| MAX\_CATEGORIE | Costante | Numero massimo di categorie che si possono avere nel sistema | 50 |
| MAX\_PREFERENZE | Costante | Numero massimo di preferenze che si possono avere nel sistema | 500000 |
| MAX\_PRODOTTI | Costante | Numero massimo di prodotti che si possono avere nel sistema | 2000 |
| MAX\_UTENTI | Costante | Numero massimo di utenti che si possono avere nel sistema | 1000 |
| N\_PRODOTTI\_ACQUISTI | Costante | Indica quanti elementi vengono visualizzati per la lista 'Prodotti piu' acquistati' | 20 |
| N\_PRODOTTI\_NUOVI | Costante | Indica quanti elementi vengono visualizzati per la lista 'Prodotti nuovi' | 20 |
| N\_UTENTI\_ACQUISTI | Costante | Indica quanti elementi vengono visualizzati per la lista '[Utenti](file:///C:\Users\Amedeo%20Racanati\workspace\LittleAmazon\html\struct_utente.html) che effettuano piu' acquisti' | 20 |
| N\_UTENTI\_NUOVI | Costante | Indica quanti elementi vengono visualizzati per la lista 'Utenti nuovi' | 20 |
| NOT\_FOUND | Costante | E' utilizzato nella ricerca dell'indice di un elemento all'interno dell'array, e indica che non è stato trovato l'elemento | -1 |
| TASTO\_ESC | Costante | Tasto "esc", numero corrispondente quando viene premuto | 27 |
| TASTO\_INVIO | Costante | Tasto ‘invio’, numero corrispondente quando viene premuto | 13 |
| PREFERENZE\_UTENTE | Costante | Numero che indica una media delle preferenze che un utente puo' avere | 200 |
| PRODOTTI\_DA\_RACCOMANDARE | Costante | Numero dei prodotti da raccomandare | 4 |
| STIMA\_ACQUISTATO | Costante | Valore di stima di un prodotto da raccomandare se acquistato da utenti affini | 1.5 |
| STIMA\_LISTA\_DESIDERI | Costante | Valore di stima di un prodotto da raccomandare se aggiunto alla lista desideri da utenti affini | 1 |
| STIMA\_VISUALIZZATO | Costante | Valore di stima di un prodotto da raccomandare se visualizzato da utenti affini | 0.5 |
| UTENTI\_SIMILI\_DA\_CONSIDERARE | Costante | Numero che indica quanti utenti simili vengono considerati per la ricerca dei prodotti raccomandati | 5 |
| TIPO\_PREFERENZA\_LISTA\_DESIDERI | Costante | Indica che il prodotto e' stato aggiunto alla lista dei desideri | 0 |
| TIPO\_PREFERENZA\_VISUALIZZATO | Costante | Indica che il prodotto e' stato visualizzato | .1 |
| FILE\_UTENTI | Costante | Nome del file contenente i dati degli utenti | Utenti.txt |
| FILE\_PREFERENZE | Costante | Nome del file contenente i dati delle preferenze | Preferenze.txt |
| FILE\_CATEGORIE | Costante | Nome del file contenente i dati delle categorie | Categorie.txt |
| FILE\_PRODOTTI | Costante | Nome del file contenente i dati dei prodotti | Prodotti.txt |
| Categorie.txt | File | File ad accesso binario contenente i dati delle categorie | Le categorie salvate |
| Prodotti.txt | File | File ad accesso binario contenente i dati i prodotti | I prodotti salvati |
| Utenti.txt | File | File ad accesso binario contenente i dati degli utenti | Gli utenti salvati |
| Preferenze.txt | File | File ad accesso binario contenente i dati delle preferenze | Le preferenze salvate |

**PROGETTAZIONE DELLE LIBRERIE / FUNZIONI**

Elenco delle librerie: Utility.h, Preferenze.h, Utenti.h, Prodotti.h, Categorie.h

Ogni libreria gestisce un tipo di dato specifico (utente, prodotto,…), tranne la libreria Utility.h, che contiene funzioni di carattere generale utilizzate dalle altre librerie.

Per maggiori informazioni si veda la documentazione prodotta con Doxygen.

Ora vengono elencate le funzioni di ciascuna libreria:

**Utility.h**

**void** **GestisciMenu**(MENU menu[], **int** NumeroElementi);

**void** **SetColor**(**int** ForgC);

**void** **CopyArray**(**char** arrTo[], **char** arrFrom[], **int** length);

**void** **GetStringFromInput**(**char** str[], **int** length);

**void** **GetStringFromInputExact**(**char** str[], **int** length, bool alsoNullable);

**float** **GetFloatFromInput**(bool strettamentePositivo);

**int** **GetIntegerFromInput**(**int** min, **int** max);

**void** **GetDataFromInput**(Data \*dataPtr);

bool **IsFirstDataBigger**(Data data1, Data data2);

**char**\* **GetLowerString**(**char** str[], **const** **int** length);

**void** **OrdinaArrayPreferenze**();

**Preferenze.h**

**void** **GestionePreferenze**();

**int** **GetSceltaPreferenza**(**int** numeroPezzi, **int** IDUtente, **int** IDProdotto);

**int** **GetAcquistiUtente**(**int** IDUtente);

**int** **GetVenditeProdotto**(**int** IDProdotto);

**int** **GetIndexPreferenzeFromArray**(**int** IDUtente, **int** IDProdotto);

**Utenti.h**

**void** **UtenteRicerca**();

**void** **UtenteInserimento**();

**void** **UtenteVisualizza**();

**void** **UtenteVisualizzaIntro**(Utente utente);

**void** **UtenteVisualizzaPreferenze**(**int** IDUtente);

**void** **UtenteModifica**(Utente \*utente, **int** index);

**void** **UtenteElimina**(**int** index, Utente utente);

**void** **UtenteElencoAcquisti**();

**void** **UtenteElencoNuovi**();

**void** **ShowUtenteForElenco**(Utente utente);

bool **ExistsOneUtente**();

**int** **GetIndexUtenteFromArrayByID**(**int** ID);

**void** **GetProdottiRaccomandati**(**int** IDUtente);

**void** **GetUtenteSimilarita**(Preferenza preferenzeUtentePrincipale[], **int** nPrefUtentePrincipale, UtenteSimilarita \*utente2);

**int** **GetElencoPreferenzeUtente**(**int** IDUtente, Preferenza preferenze[]);

**void** **EliminazionePreferenzeUtente**(**int** IDUtente);

**Prodotti.h**

**void** **ProdottoRicerca**();

**void** **ProdottoInserimento**();

**void** **ProdottoVisualizza**();

**void** **ProdottoVisualizzaIntro**(Prodotto prodotto);

**void** **ProdottoModifica**(Prodotto \*prodotto, **int** index);

**void** **ProdottoElimina**(Prodotto prodotto, **int** index);

**void** **ProdottoRicercaDitta**();

**void** **ProdottoRicercaPrezzo**();

**void** **ProdottoElencoNuovi**();

**void** **ProdottoElencoPiuVenduti**();

**void** **ShowProdottoForElenco**(Prodotto prodotto);

bool **ExistsOneProdotto**();

**int** **GetIndexProdottoFromArrayByID**(**int** ID);

**Categorie.h**

**void** **CategoriaElenco**();

**void** **CategoriaInserimento**();

**void** **CategoriaEliminazione**();

bool **CheckCategoria**(Categoria categoria);

bool **ExistsOneCategoria**();

**DIPENDENZE TRA FUNZIONI**

Ora verranno elencate tutte quelle funzioni che hanno dipendenze verso altre funzioni.

**Utility.h**

**void** **GestisciMenu**(MENU menu[], **int** NumeroElementi);

Richiama: SetColor

**void** **SetColor**(**int** ForgC);

**void** **CopyArray**(**char** arrTo[], **char** arrFrom[], **int** length);

**void** **GetStringFromInput**(**char** str[], **int** length);

**void** **GetStringFromInputExact**(**char** str[], **int** length, bool alsoNullable);

**float** **GetFloatFromInput**(bool strettamentePositivo);

**int** **GetIntegerFromInput**(**int** min, **int** max);

**void** **GetDataFromInput**(Data \*dataPtr);

Richiama: GetIntegerFromInput

bool **IsFirstDataBigger**(Data data1, Data data2);

**char**\* **GetLowerString**(**char** str[], **const** **int** length);

Richiama: CopyArray

**void** **OrdinaArrayPreferenze**();

**Preferenze.h**

**void** **GestionePreferenze**();

Richiama: ExistsOneProdotto, ExistsOneUtente, GetIndexProdottoFromArrayByID, GetIndexUtenteFromArrayByID, GetIntegerFromInput, GetSceltaPreferenza

**int** **GetSceltaPreferenza**(**int** numeroPezzi, **int** IDUtente, **int** IDProdotto);

Richiama: GetIndexPreferenzeFromArray, GetIntegerFromInput, OrdinaArrayPreferenze

**int** **GetAcquistiUtente**(**int** IDUtente);

**int** **GetVenditeProdotto**(**int** IDProdotto);

**int** **GetIndexPreferenzeFromArray**(**int** IDUtente, **int** IDProdotto);

**Utenti.h**

**void** **UtenteRicerca**();

Richiama: ExistsOneUtente, GetLowerString, GetStringFromInput, ShowUtenteForElenco

**void** **UtenteInserimento**();

Richiama: GetDataFromInput, GetIntegerFromInput, GetStringFromInput, GetStringFromInputExact

**void** **UtenteVisualizza**();

Richiama: EliminazionePreferenzeUtente, GetIndexUtenteFromArrayByID, GetIntegerFromInput, GetProdottiRaccomandati, UtenteElimina, UtenteModifica, UtenteVisualizzaIntro, UtenteVisualizzaPreferenze

**void** **UtenteVisualizzaIntro**(Utente utente);

**void** **UtenteVisualizzaPreferenze**(**int** IDUtente);

Richiama: GetIndexProdottFromArrayByID

**void** **UtenteModifica**(Utente \*utente, **int** index);

Richiama: GetDataFromInput, GetIntegerFromInput, GetStringFromInput, GetStringFromInputExact

**void** **UtenteElimina**(**int** index, Utente utente);

**void** **UtenteElencoAcquisti**();

Richiama: ExistsOneUtente, GetAcquistiUtente

**void** **UtenteElencoNuovi**();

Richiama: ExistsOneUtente, IsFirstDataBigger

**void** **ShowUtenteForElenco**(Utente utente);

bool **ExistsOneUtente**();

**int** **GetIndexUtenteFromArrayByID**(**int** ID);

**void** **GetProdottiRaccomandati**(**int** IDUtente);

Richiama: GetElencoPreferenzeUtente, GetIndexProdottoFromArrayByID, GetUtenteSimilarita, ShowProdottoForElenco

**void** **GetUtenteSimilarita**(Preferenza preferenzeUtentePrincipale[], **int** nPrefUtentePrincipale, UtenteSimilarita \*utente2);

Richiama: GetELencoPreferenzeUtente

**int** **GetElencoPreferenzeUtente**(**int** IDUtente, Preferenza preferenze[]);

**void** **EliminazionePreferenzeUtente**(**int** IDUtente);

Richiama: OrdinaArrayPreferenze

**Prodotti.h**

**void** **ProdottoRicerca**();

Richiama: ExistsOneProdotto, GetLowerString, GetStringFromInput, ShowProdottoForElenco

**void** **ProdottoInserimento**();

Richiama: CheckCategoria, ExistsOneCategoria, GetDataFromInput, GetFloatFromInput, GetIntegerFromInput, GetStringFromInput

**void** **ProdottoVisualizza**();

Richiama: GetIndexProdottoFromArrayByID, GetIntegerFromInput, OrdinaArrayPreferenze, ProdottoElimina, ProdottoModifica, ProdottoVisualizzaIntro

**void** **ProdottoVisualizzaIntro**(Prodotto prodotto);

**void** **ProdottoModifica**(Prodotto \*prodotto, **int** index);

Richiama: CheckCategoria, GetDataFromInput, GetFloatFromInput, GetIntegerFromInput, GetStringFromInput

**void** **ProdottoElimina**(Prodotto prodotto, **int** index);

**void** **ProdottoRicercaDitta**();

Richiama: ExistsOneProdotto, GetLowerString, GetStringFromInput, ShowProdottoForElenco

**void** **ProdottoRicercaPrezzo**();

Richiama: ExistsOneProdotto, GetFloatFromInput, ShowProdottoForElenco

**void** **ProdottoElencoNuovi**();

Richiama: ExistsOneProdotto, IsFirstDataBigger

**void** **ProdottoElencoPiuVenduti**();

Richiama: ExistsOneProdotto, GetVenditeProdotto

**void** **ShowProdottoForElenco**(Prodotto prodotto);

bool **ExistsOneProdotto**();

**int** **GetIndexProdottoFromArrayByID**(**int** ID);

**Categorie.h**

**void** **CategoriaElenco**();

**void** **CategoriaInserimento**();

Richiama: CheckCategoria, CopyArray

**void** **CategoriaEliminazione**();

Richiama: CopyArray

bool **CheckCategoria**(Categoria categoria);

bool **ExistsOneCategoria**();

**LittleAmazon.c**

bool **CaricamentoUtenti**();

bool **CaricamentoProdotti**();

bool **CaricamentoCategorie**();

Richiama: CopyArray

bool **CaricamentoPreferenze**();

**void** **GestioneProdotti**();

Richiama: GestisciMenu, ProdottoRicerca, ProdottoInserimento, ProdottoVisualizza, ProdottoRicercaDitta, ProdottoRicercaPrezzo, ProdottoElencoNuovi, ProdottoElencoPiuVenduti

**void** **GestioneCategorie**();

Richiama: GestisciMenu, CategoriaElenco, CategoriaInserimento. CategoriaEliminazione

**void** **GestioneUtenti**();

Richiama: GestisciMenu, UtenteRicerca, UtenteInserimento, UtenteVisualizza, UtenteElencoAcquisti, UtenteElencoNuovi

**void** **DisposeCategorie**();

**void** **DisposeProdotti**();

**void** **DisposeUtenti**();

**void** **DisposePreferenze**();

**FLOW-CHART**

I flow chart delle principali funzioni sono presenti nella stessa cartella Flow-chart, in formato pdf.

Le funzioni elencate riguardano:

* l’algoritmo di ricerca dei prodotti raccomandati;
* l’ordinamento dell’array delle preferenze;
* eliminazione delle preferenze di un utente;
* visualizzazione dei dati di un utente;
* elenco dei prodotti nuovi;
* elenco dei prodotti di una determinata fascia di prezzo.