1506
UNIVERSITÀ
DEGLI STUDI
DI URBINO
CARLO BO

CORSO DI LAUREA IN
INFORMATICA APPLICATA
SCUOLA DI
SCIENZE TECNOLOGIE E FILOSOFIA DELL'INFORMAZIONE

STUDENTE: Edoardo Gamurrini

MATRICOLA: 293843

CORSO DI PROGRAMMAZIONE E MODELLAZIONE A OGGETTI

PROGETTO SOFTWARE PER LA SESSIONE DI ESAME AUTUNNALE 2020 – 2021

Docente: Saverio Del Priori

SPECIFICA DEL SOFTWARE

Il progetto prevede un software grafico utilizzato da un ipotetico circolo tennis, con la capacità di gestire la lista dei suoi soci. In particolare, il programma permette di:

- Aggiungere un nuovo socio;
- Rimuovere un socio esistente;
- Modificare i dati relativi ad un socio;
- Aggiornare con relativa data di modifica, la classifica del socio;
- Consultare tutte le caratteristiche del socio raccolte in una scheda specifica;
- Visualizzare un grafico che rappresenta l'andamento della classifica del socio;
- Salvare il database contenente tutti i soci in un file JSON.

Per ogni socio sono memorizzati i dati che seguono:

- Nome;
- Cognome;
- Sesso;
- Età;
- Altezza;
- Classifica.

STUDIO DEL PROBLEMA

Punti critici che costituiscono il progetto:

- Aggiungere un nuovo socio: viene richiesto all'utente di inserire i dati del soggetto che si vuole aggiungere quali nome, cognome, data di nascita, sesso, classifica ed altezza espressa in metri. Per far sì che il programma crei un nuovo elemento da aggiungere alla lista ListView, l'utente deve cliccare l'apposito tasto "OK" presente nell'apposita interfaccia. È presente un tasto "Annulla" per tornare alla schermata principale;
- Rimuovere un socio esistente: viene richiesto all'utente di selezionare il soggetto che si vuole eliminare dalla lista e confermare premendo l'apposito tasto "Sì" presente nel messaggio di avvertenza che viene automaticamente aperto. Nel caso in cui non venga selezionato alcun socio prima di procedere con l'eliminazione o venga premuto il tasto "No" all'apertura del messaggio, non viene rimosso alcun socio e si torna alla pagina iniziale;
- Modificare i dati relativi ad un socio: viene richiesto all'utente di selezionare il soggetto che si vuole modificare, come nel precedente caso, dalla lista e di seguito l'apposito tasto per la modifica. Se non viene seguito l'ordine descritto, viene riportato un messaggio di errore. Apertasi la schermata, l'utente può modificare i vari dati del socio ed una volta premuto il tasto "OK" il programma si occuperà di eliminare il soggetto per poi aggiungerlo nella stessa posizione della lista ma con i dati aggiornati. È presente un tasto "Annulla" per tornare alla schermata principale;
- Aggiornare la classifica del socio: viene nuovamente richiesta la selezione del soggetto al quale si vuole aggiungere la classifica ed il tasto apposito per l'apertura della finestra, se non si vuole visualizzare l'errore. Apertasi la schermata, l'utente ha la possibilità di aggiungere una classifica e la relativa data in cui si presuppone stia utilizzando il software (quindi necessariamente successiva all'ultima data di inserimento). A questo punto cliccando il tasto "OK" il programma si occuperà di aggiungere la data nella "datelist" e la classifica nella "rankinglist". È presente un tasto "Annulla" per tornare alla schermata principale;

- Consultare tutte le caratteristiche del socio raccolte in una scheda specifica: anche in questo caso è richiesta la selezione del soggetto del quale si vogliono visualizzare i dati e dell'apposito tasto, se non si vuole riscontrare un errore. Fatto ciò, l'utente si troverà una scheda di sola consultazione dove potrà comodamente visionare l'andamento del socio tramite i suoi dati. Nello specifico sono riportati nome, cognome, sesso, età, altezza, classifica iniziale, classifica attuale, le posizioni guadagnate o perse e la classifica media;
- Salvare il database contenente tutti i soci: il caricamento ed il salvataggio dei soci e dei relativi dati è compito del programma il quale si avvale di un file JSON che viene caricato quando si avvia il software e viene salvato durante la chiusura dello stesso.

SCELTE ARCHITETTURALI

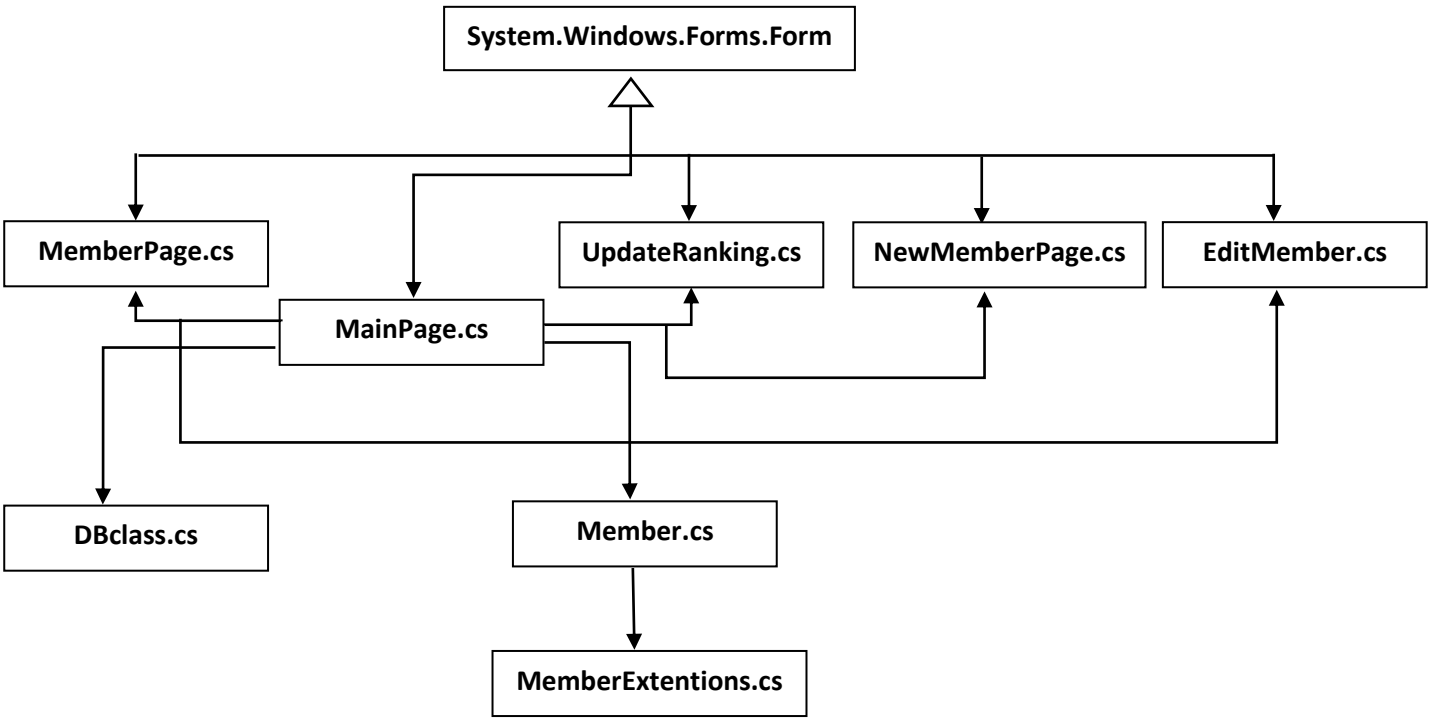
Il programma è suddiviso in varie classi. La principale che domina le altre, porta il nome di "MainPage.cs". All'interno della sua form si trovano i vari bottoni per il collegamento delle classi subordinate ad essa, nella parte superiore, mentre inferiormente è possibile visualizzare una tabella (la listview) dove viene riportato l'elenco dei soci del circolo tennis con i relativi dati ed un grafico raffigurante l'andamento della classifica del socio, il quale comparirà nel momento in cui si seleziona un socio dalla lista. Le altre classi presenti sono:

- "EditMemberPage.cs", form per la modifica dei dati relativi ad un socio;
- "MemberPage.cs", la scheda di riepilogo dei dati relativi al socio selezionato;
- "NewMemberPage.cs", form per aggiungere un nuovo socio ed i suoi relativi dati;
- "UpdateRankingPage.cs", form dove si può aggiornare la classifica del socio.

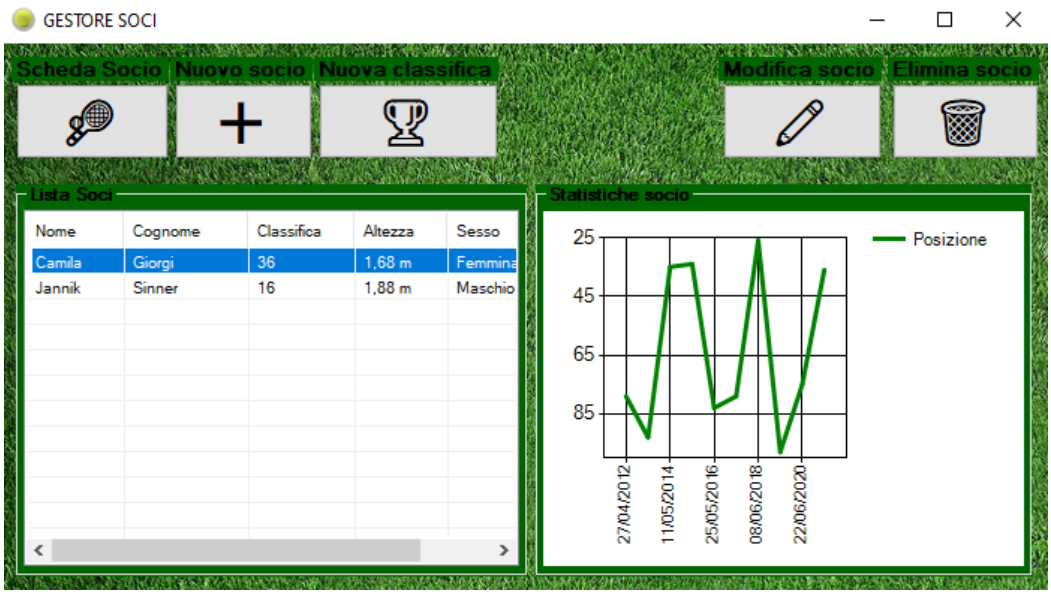
Oltre ad esse sono presenti le classi prive di form:

- "Member.cs", dove vengono riportate tutte le informazioni relative al socio;
- "MemberExtentions.cs", serve per creare il nuovo oggetto della listview;
- "DBclass.cs", come riporta il nome utile per la gestione del database dove sono contenuti tutti i soci.

Diagramma UML



MainPage.cs



La pagina mostra il menu principale del programma mettendo a disposizione dell'utente un'interfaccia semplice divisa in due parti. Nella parte superiore sono presenti i cinque bottoni per accedere alle altre form, mentre nella parte inferiore sono visibili a sinistra e a destra rispettivamente il database contenente tutti i soci memorizzati dal circolo con i loro dati ed un grafico raffigurante l'andamento della classifica del socio selezionato. La selezione del socio prima di cliccare uno dei bottoni è richiesta necessariamente se non si vuole visualizzare un messaggio di errore.

Metodi della classe

- **bttnMemberPage_Click:** porta il nome del bottone per visualizzare la scheda riassuntiva dei dati del socio. Se non si vuole visualizzare un messaggio di errore è richiesta la preventiva selezione del socio interessato, fatto ciò, alla pressione del tasto il programma apre la form relativa di tipo "MemberPage" e le passa come parametro il socio selezionato;
- **bttnNewMember_Click:** porta il nome del bottone per creare un nuovo socio. Rende possibile l'apertura della form interessata, "NewMemberPage" e le passa la lista dei soci come parametro per poi, alla pressione del tasto "OK", inserire il nuovo socio creato e tramutato in elemento della lista, all'interno della listview. Se si preme il tasto "Annulla" invece, la lista rimarrà invariata dopo aver fatto un controllo sull'attributo di tipo index;
- **bttnUpdateRanking_Click:** porta il nome del bottone per aggiornare la classifica di un socio. Se non si vuole visualizzare un messaggio di errore è richiesta la preventiva selezione del socio interessato, fatto ciò, alla pressione del tasto il programma apre la form relativa di tipo "UpgradeRankingPage" e le passa come parametro il socio selezionato. Aggiunta la nuova classifica e la data relativa, il programma elimina il socio precedente e inserisce quello modificato nella stessa posizione della listview;
- **bttnEditMember_Click:** porta il nome del bottone per modificare il socio. Se non si vuole visualizzare un messaggio di errore è richiesta la preventiva selezione del socio interessato, fatto ciò, alla pressione del tasto il programma apre la form relativa di tipo "EditMemberPage" e le passa come parametro il socio selezionato. Apportate le dovute modifiche, il programma elimina il socio precedente e inserisce quello modificato nella stessa posizione della listview;
- **bttnDelete_Click:** porta il nome del bottone per eliminare il socio. Se non si vuole visualizzare un messaggio di errore è richiesta la preventiva selezione del socio interessato, fatto ciò, viene visualizzato un messaggio di avviso, per dare la possibilità all'utente di annullare l'azione. A questo

punto se si clicca su “sì” si rimuove il socio ed il relativo grafico. La rimozione del socio aggiorna anche l’index in modo tale che il prossimo socio venga inserito nella posizione corretta;

- **LoadDB:** richiama semplicemente il metodo del dbHandler per caricare la lista dei soci nella listview. Questo metodo viene eseguito una volta avviato il programma;
- **lstVwMain_ItemActivate:** quando un elemento della listview viene selezionato, questo metodo viene invocato ed entra in azione eliminando il grafico precedente e creandone uno nuovo relativo al socio che è stato selezionato visitando la lista di date e classifiche. Per la costruzione del grafico, risulta utile calcolare il valore minimo della classifica settandolo come valore di base dell’asse delle Y, che poi si trasformerà in valore massimo, dato che è stato scelto di visualizzare questo asse al contrario per fare in modo che risulti più veritiera la “scalata” della classifica compiuta da un socio e non sia una discesa;
- **FinalSaveData:** richiama semplicemente il metodo del dbHandler per salvare la lista dei soci nel file JSON. Questo metodo viene eseguito una volta chiuso il programma.

Attributi della classe

- Elemento int **index**, per la gestione di inserimenti e rimozioni all’interno della lista di soci;
- Lista **members**, contenente elementi di tipo Member, nella quale sono appunto memorizzati tutti i soci del circolo tennis. Viene dichiarata private perché utilizzata solo in questa classe o passata alle altre forms;
- Oggetto **memberItem** che serve per aggiungere i soci alla lista esistente listview, infatti è di tipo ListViewItem come tutti gli altri componenti;
- Oggetto **dbHandler**, come dice il nome stesso serve per la gestione del database, nel dettaglio, per il caricamento ed il salvataggio di tutti i dati dei soci presenti nel file JSON.

MemberPage.cs

La pagina mostra la scheda raffigurante tutti i dati del socio selezionato e le sue relative statistiche riguardanti l’andamento della classifica. Non sono presenti bottoni perché in questo caso si tratta di una form creata esclusivamente per la consultazione.

Metodi della classe

- **RankingChangedCalculator:** calcola la differenza delle posizioni del socio selezionato, tra la prima classifica inserita e la attuale posizione;
- **AverageRankingCalculator:** calcola la classifica media tra tutte quelle memorizzate.

Attributi della classe

- Oggetto private **memberStats** di tipo Member utilizzato solamente all'interno di questa classe.

NewMemberPage.cs

La pagina per creare un nuovo socio del circolo tennis dove è richiesto di inserire i dati relativi ad esso.

Metodi della classe

- **AddMember:** utilizzato per creare un nuovo socio quindi un nuovo oggetto di tipo Member. Questo avrà al suo interno tutti dati inseriti dall'utente e sarà inserito da questo metodo all'interno della lista di soci. Tra le specifiche troviamo anche la data di inserimento;
- **bttnOk_Click:** metodo richiamato al click sul tasto OK presente nella form il quale richiama al suo interno i metodi "CheckFields" per il controllo del riempimento di tutti i dati richiesti. Passa poi al richiamo del metodo "CheckRadioButton", solo se il precedente è stato soddisfatto, per quanto concerne il controllo di quale sesso sia stato inserito. Infine, richiama "AddMember" spiegato precedentemente e chiude la form. Viene visualizzato un errore se non tutti i campi sono stati correttamente riempiti;
- **bttnCancel_Click:** richiamato quando viene cliccato il tasto annulla presente nella form. Non fa altro che chiudere la form e lasciare tutto invariato;
- **CheckFields:** come spiegato precedentemente, si occupa del controllo che tutti i campi siano stati riempiti restituendo true se soddisfatto e false altrimenti;
- **CheckRadioBttn:** come spiegato precedentemente, si occupa del controllo di quale bottone sia stato selezionato tra Maschio e Femmina. In base a quale sesso sia stato selezionato viene settata la variabile apposita sex;
- **HadleInput:** metodo per il controllo che l'utente non inserisca input sbagliati.

Attributi della classe

- Variabile private String **sex**, per settare il sesso del socio e per il radio button;
- Lista public **members** ti tipo Member utilizzata anche in altre classi.

UpdateRankingPage.cs

La pagina mostra nella parte superiore lo spazio per l'inserimento della data in cui si sta aggiornando la classifica del socio selezionato precedentemente. La data può essere solamente prossima a quella dell'ultimo inserimento. Nella parte inferiore si trova lo spazio per specificare il nuovo valore della classifica.

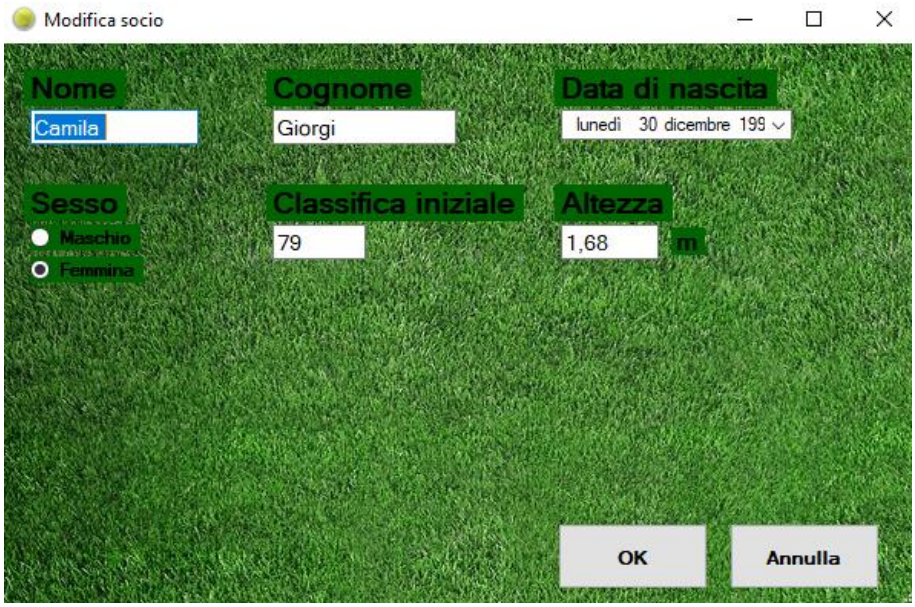
Metodi della classe

- **bttnOk_Click:** richiamato al click del pulsante OK presente nella form, aggiunge la data e la nuova classifica nelle liste del socio interessato e chiude la form;
- **bttnCancel_Click:** richiamato quando viene cliccato il tasto annulla presente nella form. Non fa altro che chiudere la form e lasciare tutto invariato;
- **HandleInput:** metodo per il controllo che l'utente non inserisca input sbagliati.

Attributi della classe

- Oggetto public **member** di tipo member, utilizzato in questa classe e non solo.

EditMemberPage.cs



Metodi della classe

- **SetRadioButton:** metodo per avere già pronto e salvato il sesso del socio. Quando viene aperta la form, grazie a questo metodo, la spunta sul sesso è già settata;
- **bttnOk_Click:** richiamato al click del pulsante OK presente nella form, aggiunge i nuovi valori inseriti nell'oggetto editedMember;
- **bttnCancel_Click:** richiamato quando viene cliccato il tasto annulla presente nella form. Non fa altro che chiudere la form e lasciare tutto invariato;
- **HandleInput:** metodo per il controllo che l'utente non inserisca input sbagliati.

Attributi della classe

- Oggetto public **editedMember** di tipo Member, utilizzato in questa classe e non solo.

DBclass.cs

Questa classe è priva di interfaccia grafica e serve alla gestione del database da come richiama il nome stesso per gestire il database. Nello specifico in essa vengono implementati i metodi per salvare tutti i dati relativi ai soci del circolo tennis in un file JSON e viceversa.

La particolarità di questa classe in confronto alle altre è l'implementazione del pattern Singleton. È stato scelto questo per fare in modo di avere una sola istanza di un oggetto ed evitare ricorrenze di oggetti identici.

Metodi della classe

- **GetData:** necessaria l'installazione del pacchetto "Newtonsoft.Json". Il metodo è usato per la lettura dei dati presenti nel file JSON. Una volta memorizzati vengono resi oggetti di tipo Member;
- **SaveData:** necessaria l'installazione del pacchetto "System.Text.Json". Il metodo è usato per il salvataggio dei soci all'interno del file JSON;
- **GetIstance:** è il metodo del pattern singleton, come spiegato precedentemente questo può essere richiamato quando si vuole creare un oggetto di tipo DBclass ed essere sicuri che sia invariato.

Attributi della classe

- Attributo di tipo DBclass **istance**, utilizzato per l'implementazione del pattern singleton.

Member.cs

Classe dell'oggetto di tipo Socio nella quale sono raccolte tutte le varie caratteristiche del socio stesso, come ad esempio il nome, il cognome, la classifica ed altre.

Metodi della classe

- **AgeCalculator:** utilizzato per il calcolo dell'età del socio;
- **LastRanking:** utilizzato per la ricerca e la restituzione dell'ultima classifica inserita nella lista;
- **LastDate:** come il metodo precedente ma relativo all'ultima data della DateList;
- **FindMinimumRanking:** utilizzato per la ricerca della classifica più bassa (in realtà sarebbe la più alta ma visto che si parla di numeri bisogna ricercare il più piccolo) ed è utile per la costruzione del grafico.

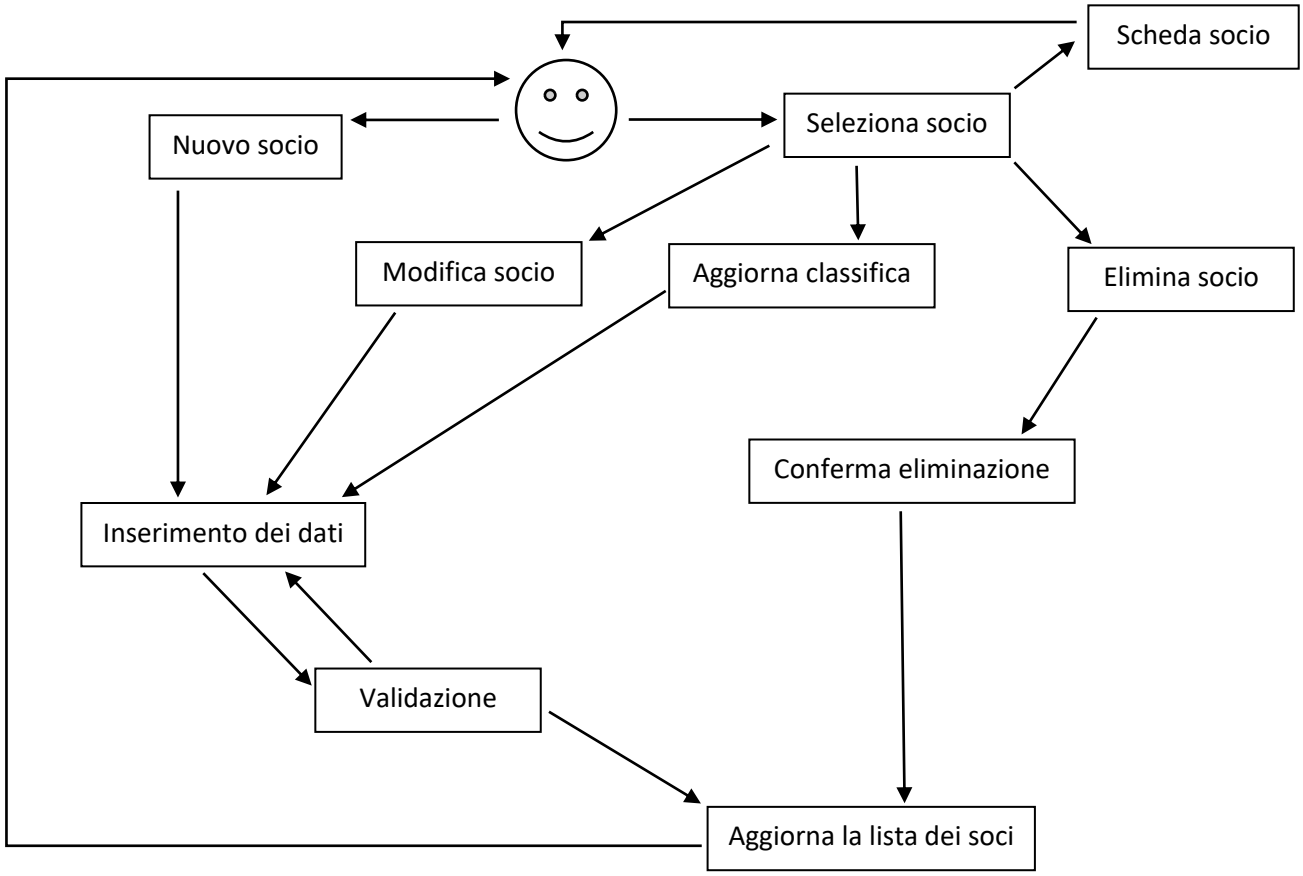
Attributi della classe

- Attributi tipo String sono **Name**, **Surname** e **Sex**, per le relative caratteristiche del socio nome, cognome e sesso;
- Attributo tipo float **Height** per quanto concerne l'altezza del socio;
- Attributo int **Age**, per la memorizzazione dell'età del socio;
- Attributo DateTime **BirthdayDate**, rappresenta la data di nascita del socio;
- Una lista tipo int **RankingLists**, la quale è utilizzata per contenere la lista di tutte le classifiche relative al socio;
- Una lista di tipo DateTime **DateList**, per la rappresentazione di tutte le date in cui è stata aggiornata la classifica del socio.

Estensioni della classe

- **MemberExtentions.cs:** viene utilizzata per la trasformazione di un oggetto da tipo Member a tipo ListViewItem per fare in modo che possa far parte della listview.

CASI D'USO



Autore	Utente
Nome Use case	Scheda socio
Numero Use case	1
Precondizione	Use case 2
Utilizzo	Click sul tasto "Scheda socio"
Postcondizione	
Alternative valide	Use case 4 or 5 or 6

Autore	Utente
Nome Use case	Selezione del socio
Numero Use case	2
Precondizione	
Utilizzo	Click sul socio per selezionarlo
Postcondizione	Use case 1 or 4 or 5 or 6
Alternative valide	Use case 3

Autore	Utente
Nome Use case	Nuovo socio
Numero Use case	3
Precondizione	
Utilizzo	Click sul tasto “Nuovo socio”
Postcondizione	Use case 7
Alternative valide	Use case 2

Autore	Utente
Nome Use case	Aggiorna classifica
Numero Use case	4
Precondizione	Use case 2
Utilizzo	Click sul tasto “Aggiorna classifica”
Postcondizione	Use case 7
Alternative valide	Use case 1 or 5 or 6

Autore	Utente
Nome Use case	Modifica socio
Numero Use case	5
Precondizione	Use case 2
Utilizzo	Click sul tasto “Modifica socio”
Postcondizione	Use case 7
Alternative valide	Use case 1 or 4 or 6

Autore	Utente
Nome Use case	Elimina socio
Numero Use case	6
Precondizione	Use case 2
Utilizzo	Click sul tasto “Elimina socio”
Postcondizione	Use case 8
Alternative valide	Use case 1 or 4 or 5

Autore	Utente
Nome Use case	Inserimento dei dati
Numero Use case	7
Precondizione	Use case 3 or 4 or 5
Utilizzo	Inserimento dei dati da parte dell’utente
Postcondizione	Use case 9
Alternative valide	

Autore	Utente
Nome Use case	Conferma eliminazione
Numero Use case	8
Precondizione	Use case 6
Utilizzo	Comparsa di un messaggio di conferma dell’eliminazione del socio
Postcondizione	Use case 10
Alternative valide	

Autore	
Nome Use case	Validazione
Numero Use case	9
Precondizione	Use case 7
Utilizzo	Verifica dei dati se sono stati inseriti correttamente
Postcondizione	Use case 10 or 7
Alternative valide	

Autore	
Nome Use case	Aggiornamento soci
Numero Use case	10
Precondizione	Use case 8 or 9
Utilizzo	Lista dei soci aggiornata
Postcondizione	
Alternative valide	