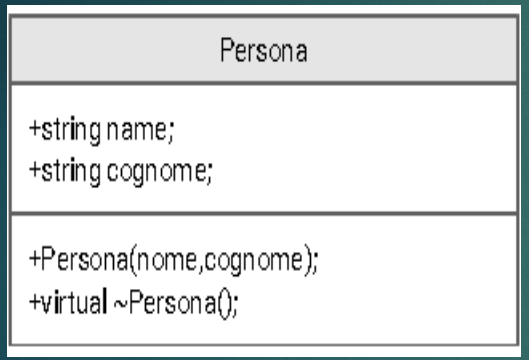
Circolo scacchistico Programmazione avanzata

JOSE RAUL LUIZAGA YUJRA 1046611

Scopo del progetto

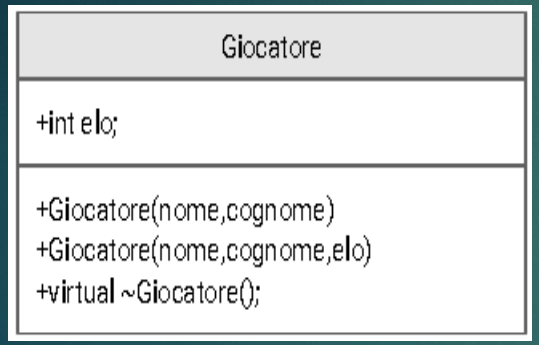
- Questo progetto propone la realizzazione di un circolo scacchistico chiamato Circolo che organizza tornei di scacchi . I giocatori possono iscriversi ad un torneo indipendentemente dall'iscrizione alla federazione scacchistica, infatti nel caso non fossero iscritti viene assegnato un elo di partenza. Come nei casi reali si propone un torneo a 5 turni con un numero minimo di 6 partecipanti, nel caso non si raggiungesse il numero pari per gli accoppiamenti delle partite tra sfidanti viene inserito un giocatore fittizio chiamato Forfait. Logica della generazione dei turni in base all'elo per il primo turno, mentre per gli altri in base al punteggio all'interno del torneo.
- Alla fine di ogni turno verrà esposta la classifica parziale del torneo.

Classe Persona



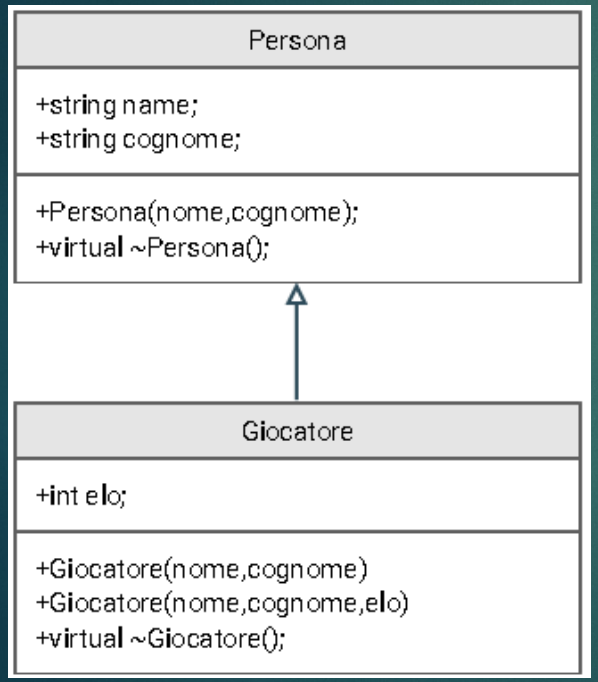
- Classe Persona contiene I campi nome e cognome che sono pubblici.
- Contiene un costruttore necessita di due campi di tipo stringa
- Distruttore virtual

Classe Giocatore



- Giocatore contiene il campo elo.
- Eredita i campi nome e cognome da persona
- Distruttore virtual

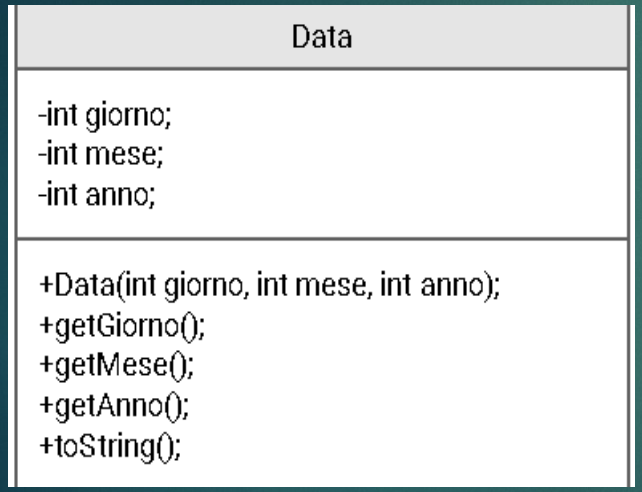
Ereditarietà tra classi Persona e Giocatore



Relazione tra le classe Persona e Giocatore:

Giocatore estende in maniera pubblica la classe Persona

Classe Data



- Classe Data contiene i campi private giorno,mese e anno.
- Contiene un costruttore necessita di tutti e 3 campi che esso contiene.
- Non ha il distruttore , nel main verrà utilizzato con un puntatore shared_ptr.
- Metodi pubblici per prendere l'informazione sulla data.

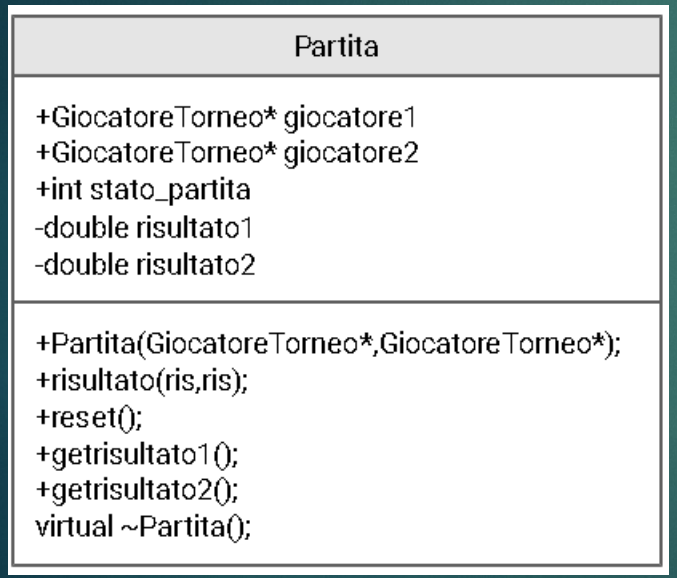
Enumerativi utilizzati

- Risultato(Vittoria,Sconfitta,Pareggio)
- Tipo(Ufficiale,NonUfficiale)

Risultato utilizzato per l'assegnamento dei punti al termine di una partita.

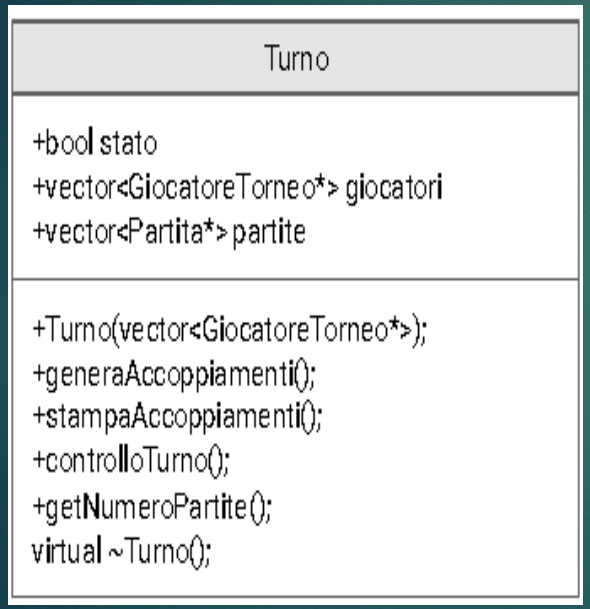
Tipo utilizzato durante la creazione di un torneo, ne consegue variazione punti elo.

Classe Partita



- Classe partita contiene 3 campi pubblici e 2 private.
- Contiene un costruttore che necessita di due campi pubblici che sono di tipo Giocatore.
- Distruttore virtual
- Contiene un metodo per inserire il risultato
- Contiene un metodo per restituire i risultati dei singoli giocatori,
- Contiene un metodo per cancellare I risultati inseriti

Classe Turno



- Classe Turno contiene 3 campi pubblici , uno per la gestione dello stato del turno ed altri due per l'amministrazione delle partite tra partecipanti al turno.
- Costruttore che prende in ingress un vettore di Giocatori al torneo.
- Distruttore virtual
- Contiene una funzione per la generazione degli accoppiamenti tra partecipanti
- Contiene altre funzioni riguardo l'informazione sul turno.

Classe Torneo

Torneo
+static const int turni = 5 -string nome; -float costo; -shared_ptr<Data> data -int codice -Tipo tipo_torneo -vector<shared_ptr<Giocatore>> lista_Iscrizioni -vector<GiocatoreTorneo*> lista_score -vector<shared_ptr<Turno>> lista_turni; bool stato_torneo = false -bool stato_turno -Turno* turno - Giocatore* forfait
+Torneo(string nome,float costo,shared_ptr<Data>,int codice, Tipo tipo_torneo); +getNome(); +getCosto(); +getData(); +getCodice(); +getTurnoComplessivo(); +inserisci(shared_ptr<Giocatore>); +avvia(); +generaTurno(); compareElo(GiocatoreTorneo*,GiocatoreTorneo*); +comparePunti(GiocatoreTorneo*,GiocatoreTorneo*); +stampaRisultati(); +ordinaElo(); +ordinaPunti(); +stampa(); +avvioTurno(); +risultatoAccoppiamento(int,int,ris,ris); +resetAccoppiamento(int,int accoppiamento); +controlloTurno(); +fineTurno(); +risultatiFinali(); +virtual ~Torneo();

Classe Torneo

- Classe Torneo contiene molti campi pubblici, tra questi i più importanti sono il vettore dei partecipanti, il vettore degli score ed il vettore dei turni.
- Il costruttore della classe torneo prende in input nome,costo,data, codice torneo ed il tipo di torneo.

- Distruttore torneo Virtual

Alcuni costrutti utilizzati dentro il progetto…

- **Pattern Singleton**: viene utilizzato per creare un unico oggetto che rappresenti il Circolo.
- **EREDITARIETÀ PUBBLICA**: tra classe Persona e Giocatore
- **TIPI ENUMERATIVI**: nella classe Torneo utilizza Tipo e la classe Partita utilizza Risultato
- **SMART POINTERS**: nel main i Giocatori, Data ed i Tornei sono stati istanziati tramite smart pointers (shared_ptr).

Output fine torneo

Classifica finale del torneo Excelsior

Nome giocatore	punteggio finale
Adi	5.0
Rob	3.5
Ema	3.0
raul	2.5
Turco	2.5
Dom	2.5
Mario	2.0
forfait	0.0