nome: string # crediti: unsigned short + Persona(nome: const string,crediti: const unsigned short) + getInfo(os: ostream&): ostream& + getNome(): string + getCrediti(): unsigned short + operator<(persona: const Persona&): bool + <<virtual>> eseguiOperazione(const Persona*): void = 0 <<friend>> operator<<(os: ostream&, persona: const Persona&): ostream&

Allenatore

- listaGiocatori: vector<Persona*>
- + Allenatore(nome: const string, crediti: const unsigned short)
- eseguiOperazione(giocatore: const Persona*: void
- + operator[] (index: const unsigned short): const Persona* const
- + unsigned short numeroAcquisti(): unsigned short
- <<friend>> operator<<(os: ostream&, allenatore: const FantaAllenatore&): ostream&</pre>

Fantacalcio

- allenatori: vector<Persona*>
- giocatoriAcquistati: vector<Persona*>
- merge(inizio: unsigned short, centrale: unsigned short, fine: unsigned short): void
- mergeSort(inizio: unsigned short, fine: unsigned short): void
- + Fantacalcio()
- + aggiungiAllenatore(allenatore: Persona*): void
- + acquistaGiocatore(allenatore: const string, giocatore Persona*): void
- + stampaAllenatori(): void
- + stampaAcquisti(nomeAllenatore: const string): void
- + stampaAcquisti(): void
- +cerca(persona: const Persona*): Persona*

+cerca(nome: const string): Persona*

Giocatore

- ruolo: char
- proprietario: const Persona*
- + Giocatore(nome: const string, prezzo: const unsigned short, ruolo: const char)
- + eseguiOperazione(prop: const Persona*): void
- + getRuolo(): char
- + getProprietario(): const Persona* const
- + operator string(): string
- + operator==(giocatore: const Giocatore&): bool
- <<friend>> operator<<(os:ostream&, const Giocatore&): ostream&
- <<friend>> operator>>(is: const istream&, Giocatore&): istream&