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

listaGiocatori: vector<Persona*>

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

Allenatore

Fantacalcio

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

Giocatore

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