model Partita - idPartita: String Giocatore - numGiocatori: int - giocatori: Giocatore[] nickname: String - stato: String - dataNascita: Date - turno: int coloreOperaio: String - eliminato: Giocatore idPartita: String // - vincitore: Giocatore - idTracciato: String // - idTracciato: String - partiteVinte: int // + Partita(giocatore): Partita + Giocatore(nickname,dataNascita,coloreOperaio): Giocatore + aggiungiGiocatore(giocatore): void + setIdPartita(): void // + getStato(): String + setIdTracciato(): void // + getNumGiocatori(): int + incrementaPartiteVinte(): void // + getGiocatorePiuGiovane(): Giocatore + azzeraPartiteVinte(): void // + getTurno(): int + getNickname(): String + getToccaA(): Giocatore + getDataNascita(): Date + getGiocatoriInGioco(): Giocatore[] + getColoreOperaio(): String + getEliminato(): Giocatore + getIdPartita(): String // + getVincitore(): Giocatore + getIdTracciato(): String // + getPartiteVinte(): int **Tracciato** - idTracciato: String - Celle: Cella[][] Operaio + Tracciato(): Tracciato + getIdTracciato(): String - numOperaio: int - proprietario: Giocatore - colore: String - posizione: Cella - cartaDivinitaAssociata: CartaDivinita + Operaio(Giocatore): Operaio + getColore(): String + getNumOperaio(): int Cella + getProprietario(): Giocatore + puoSpostarsi(): boolean - indiceRiga: int + puoQui(riga,colonna): boolean - indiceColonna: int - idTracciato: String - operaioPresente: Operaio - torre: Torre + Cella(riga,colonna): Cella + putOperaio(Operaio): void + putTorre(Torre): void + eliminaOperaio(riga,colonna): void + eliminaTorre(riga, colonna): void + ceOperaio(): boolean + ceCostruzione(): boolean + siPuoCostruire() boolean + getColoreOperaioPresente(): String + getOccupanteCella(): Operaio 0,1 <<interface>> CartaDivinita **Torre** - nome: String proprietario: Giocatore livelloTorre: int + Torre(): Torre + abilita(): + Torre(livello): Torre + getLivelloTorre(Torre): int + aggiungiLivello(Torre) void + eliminaLivello(Torre): void + torreCompleta(Torre): boolean