



Università
degli Studi di
Messina

**Dipartimento di Scienze Matematiche e Informatiche,
Scienze Fisiche e Scienze della Terra**
CORSO DI LAUREA TRIENNALE IN INFORMATICA

PROGETTO DI OBJECT ORIENTED PROGRAMMING (PROGRAMMAZIONE 2)

GIOCO DI CARTE FANTASY

Docente: **Galletta Antonino**

Studente: **Sidoti Pipirito Calogero**

Indice

Introduzione.....	pag 3
Regole e funzionalità.....	pag 4
Classi.....	pag 7
Interfacce.....	pag 15
Class Diagram.....	pag 17

Introduzione

Per questo progetto, incentrato sulla programmazione ad oggetti (OOP), ho voluto progettare ed implementare un software riguardante un gioco di carte di tipo fantasy, ispirandomi ad un noto gioco di successo per Android, Pc ed altri dispositivi, della medesima categoria.

Il linguaggio di programmazione che è stato utilizzato per realizzare ciò, è Java 8, particolarmente indicato per sfruttare in modo efficace i paradigmi della programmazione ad oggetti.

Per quanto riguarda lo strumento utilizzato per lo sviluppo, è stato utilizzato l'IDE (Integrated Development Environment) NetBeans alla sua versione 14.

Regole e funzionalità

Regole generali

- Si gioca a turno.
- I giocatori partecipanti ad ogni singola partita sono due e si sfidano tra di loro.
- I giocatori hanno un valore di vita iniziale, che può essere decrementato dagli attacchi dell'avversario.
- Vince chi resta in vita.
- Ogni giocatore pesca delle carte da un mazzo, che vanno ad aggiungersi a quelle che già possiede.
- Il mazzo è uno solo ed è in comune tra i due giocatori.
- Inizialmente, ogni giocatore ha in mano 3 carte e nessuna schierata.
- All'inizio di ogni turno, ogni giocatore pesca una carta, finché il mazzo ne possiede almeno una.
- Ogni turno, il giocatore ha a disposizione una specifica quantità di energia ("mana") che sarà decrementata, fino ad un minimo di 0, di tanto quanto è la quantità di energia ("mana") necessaria per utilizzare una determinata carta che possiede.
- La quantità di energia ("mana") ad ogni turno sarà incrementata di un punto rispetto all'inizio del turno precedente, fino ad un massimo di 10.
- Al primo turno la quantità di energia ("mana") ha un valore di 1, al secondo 2, al terzo 3, fino ad un massimo di 10.
- La quantità di energia ("mana") non utilizzata nel turno in corso viene persa e non è cumulabile per il turno successivo.
- Il giocatore possiede: carte in mano ancora da giocare, carte schierate che possono essere utilizzate per attaccare le carte schierate dell'avversario o la vita dell'avversario stessa.

Regole e funzionalità associate alle carte

- In generale, il giocatore del turno può attaccare l'avversario o le sue carte schierate, utilizzando solo le proprie carte già schierate.
- In generale, se non diversamente indicato in alcune specifiche tipologie di carte, ogni carta schierata può attaccare una sola volta per turno e dal turno successivo al quale è stata schierata.
- In generale, la carta schierata viene "distrutta" quando il suo valore di salute è uguale o minore a zero.
- Ad ogni attacco, la salute della carta che subisce l'attacco si riduce di tanto quanto è il valore di attacco della carta che ha attaccato.
- Ad ogni attacco, la salute della carta che ha attaccato si riduce di tanto quanto è il valore di attacco della carta che ha subito l'attacco.
- In ogni turno, per attaccare, il giocatore sceglie una propria carta schierata e deve poi indicare se attaccare una carta schierata dell'avversario (specificando esplicitamente quale carta) o l'avversario stesso, al fine di decrementare il valore associato alla sua vita.
- Le carte si dividono in due macrocategorie: carta di tipo "Mistica" e carta di tipo "Servitore".
 - La carta di tipo Servitore possiede due caratteristiche chiave, valore di attacco e valore di salute, attraverso le quali può attaccare o un servitore nemico schierato o direttamente l'avversario, contribuendo così a decrementare il valore associato alla sua vita, di tanto quanto è il suo valore di attacco .
 - La carta di tipo Mistica non può essere schierata per attaccare, ma una volta giocata, fornisce ai servitori schierati in quel momento, del tipo specificato nella carta Mistica in quel momento in utilizzo, un bonus di +1 al loro valore di attacco e +1 al loro valore di salute.
- Le carte di tipo Servitore si dividono in: Mago, Guerriero, Bestia, Supporto.
 - Mago: alla sua uccisione, -1 salute a un servitore nemico (casuale) schierato.
 - Guerriero: potrebbe attaccare nel turno in cui viene schierato.
 - Bestia: potrebbe attaccare due volte per turno.
 - Supporto: in ogni turno in cui è schierato, +1 salute a un servitore schierato a scelta.

- Carte Mistica:
 - Mistica Mago: una volta giocata, +1 attacco e +1 salute di ogni MAGO schierato.
 - Mistica Guerriero: una volta giocata, +1 attacco e +1 salute di ogni GUERRIERO schierato.
 - Mistica Bestia: una volta giocata, +1 attacco e +1 salute di ogni BESTIA schierato.
 - Mistica Supporto: una volta giocata, +1 attacco e +1 salute di ogni SUPPORTO schierato.

- Ogni tipo di Servitore, è a sua volta suddiviso in quattro sottotipi: Terra, Acqua, Fuoco, Elettricità.
 - Mago di tipo Terra, Mago di tipo Acqua, Mago di tipo Fuoco, Mago di tipo Elettricità.
 - Guerriero di tipo Terra, Guerriero di tipo Acqua, Guerriero di tipo Fuoco, Guerriero di tipo Elettricità.
 - Bestia di tipo Terra, Bestia di tipo Acqua, Bestia di tipo Fuoco, Bestia di tipo Elettricità.
 - Supporto di tipo Terra, Supporto di tipo Acqua, Supporto di tipo Fuoco, Supporto di tipo Elettricità.

- Ogni singolo sottotipo (Terra, Acqua, Fuoco, Elettricità) ha precedenza su altri sottotipi; cioè se, ad esempio, un servitore qualunque di tipo Terra attacca un servitore avversario qualunque di tipo Elettricità, i danni subiti dal servitore di tipo Elettricità ammontano al valore di attacco + 1 del servitore di tipo Terra che ha attaccato. Di seguito tutte le precedenze dei sottotipi:
 - Terra ha precedenza su Elettricità.
 - Acqua ha precedenza su Terra e Fuoco.
 - Fuoco ha precedenza su Terra.
 - Elettricità ha precedenza su Acqua e Fuoco.

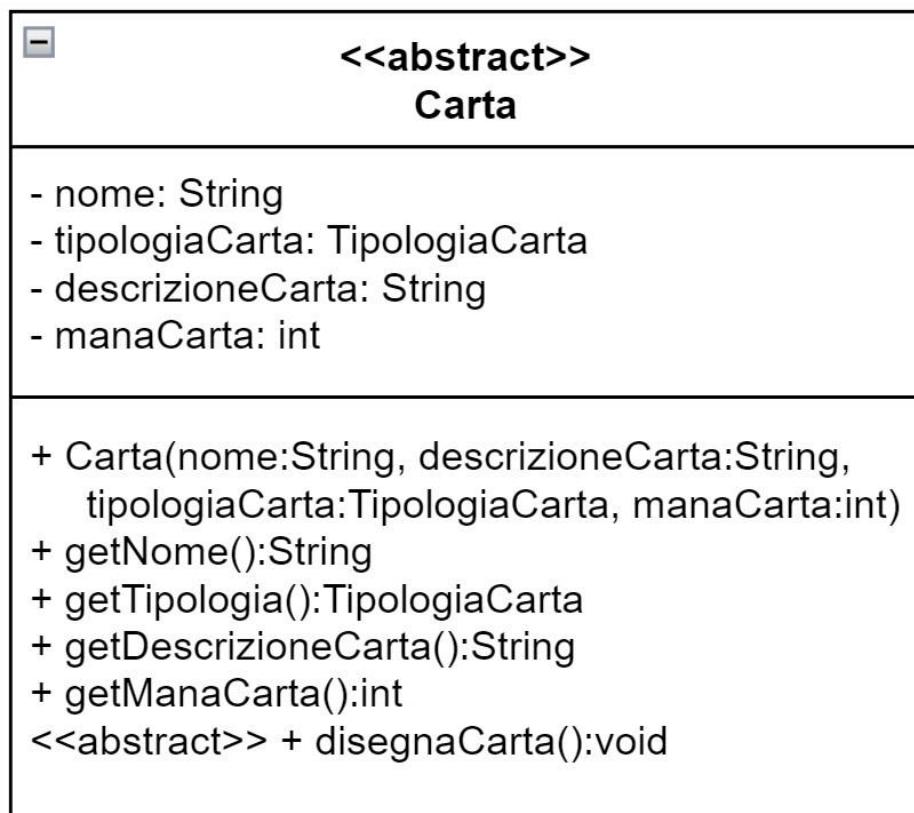
Classi

Classe Main

La classe Main è ovviamente la classe principale, responsabile dell'avvio e gestione del software mediante il metodo main() al suo interno, che richiama appunto i metodi interessati delle altre classi.

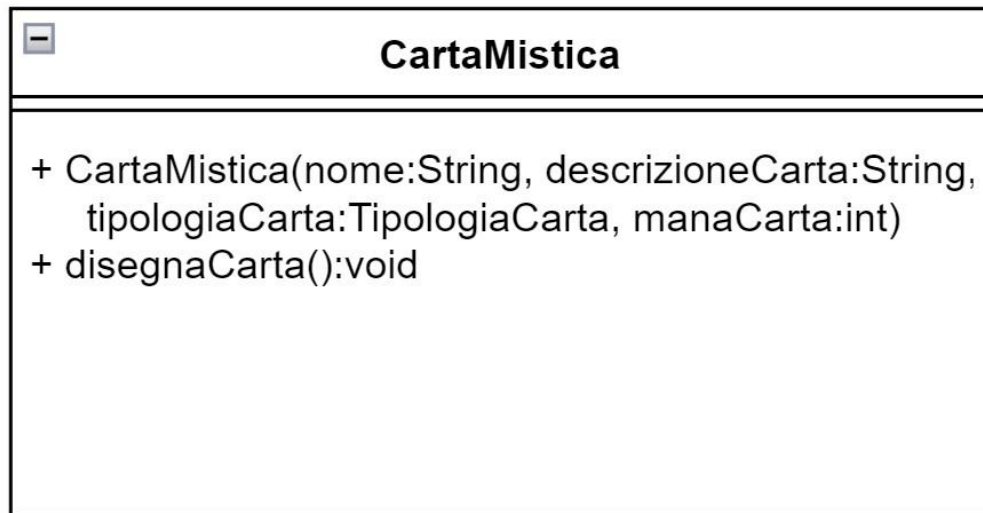
Classe Carta

E' una classe astratta, e definisce le caratteristiche base che deve avere ogni carta in gioco: nome, tipologia della carta, descrizione della carta, mana della carta necessari per essere giocata.



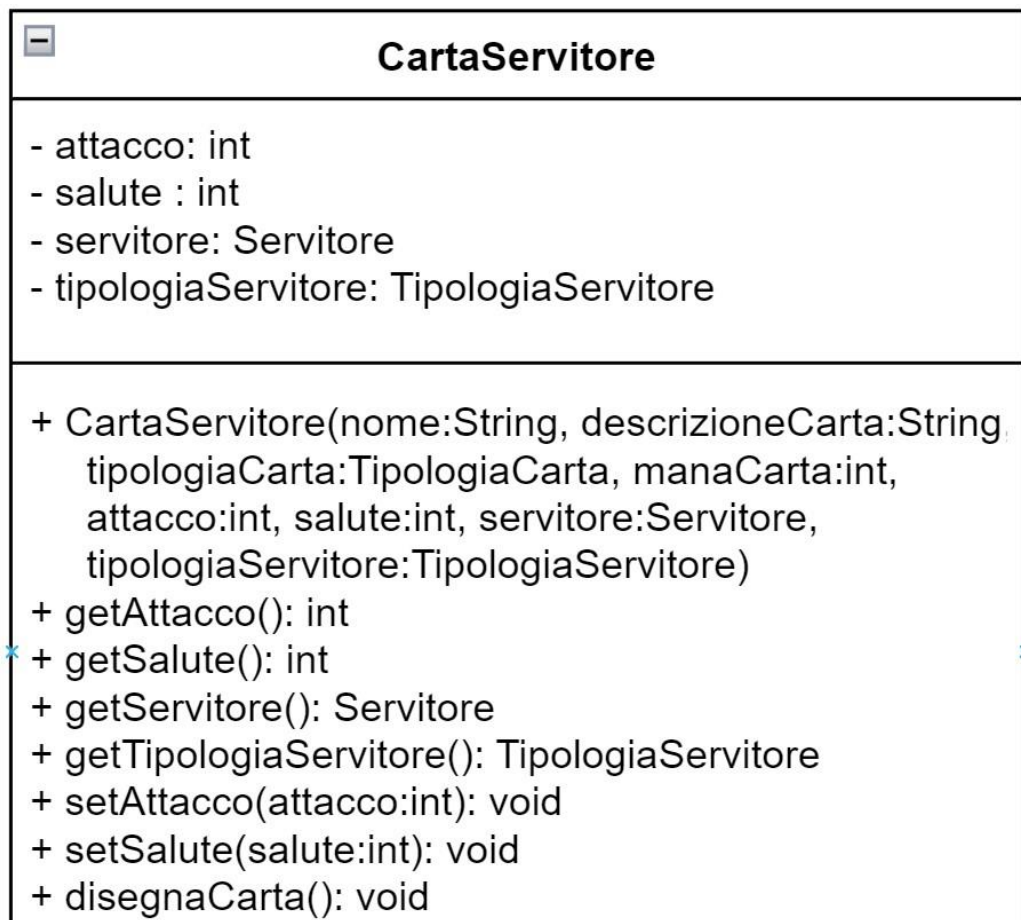
Classe CartaMistica

Questa classe estende la classe Carta. Viene fatto un override sul metodo `disegnaCarta()` della classe padre.



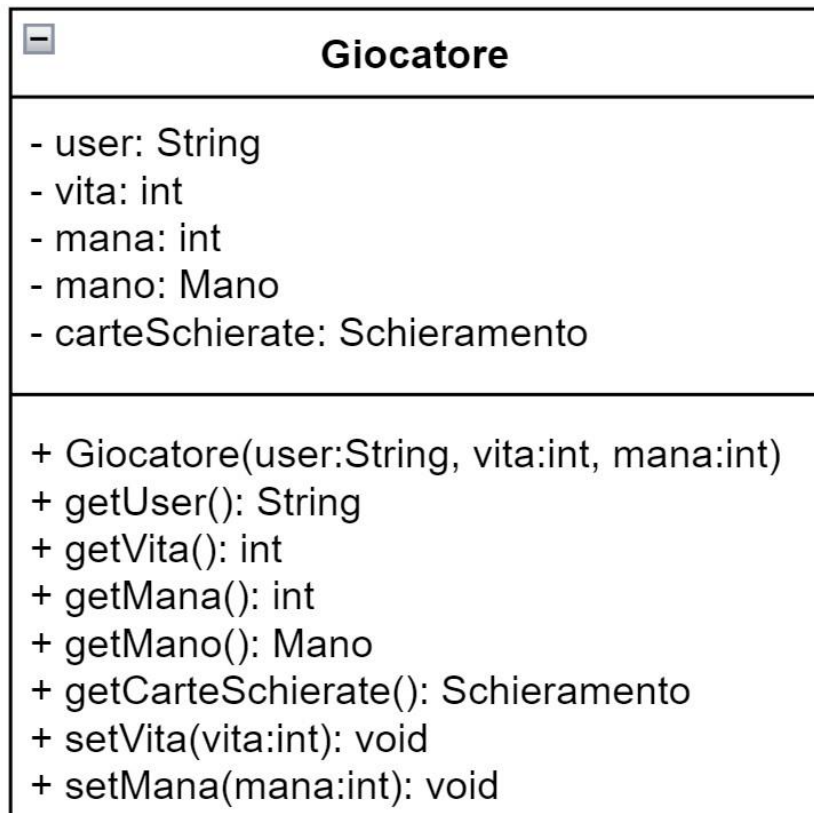
Classe CartaServitore

Questa classe estende la classe Carta. Vengono specificate le caratteristiche che deve possedere una carta di questo tipo: attacco, difesa, quale Servitore (Mago, Guerriero, Bestia, Supporto), quale tipologia (Terra, Acqua, Fuoco, Elettività). Viene fatto un override sul metodo disegnaCarta() della classe padre.




Classe Giocatore

Questa classe definisce quello che è il giocatore e le sue caratteristiche: nome, vita, mana(per ogni turno di gioco). Inoltre viene specificato nei suoi attributi che deve avere un insieme di carte che rappresenta le carte che può giocare, quindi quelle in mano, e un insieme di carte che rappresenta le carte che ha già schierato sul tavolo da gioco.




Classe Mano

Questa classe rappresenta le carte che fanno parte dell'insieme delle carte in mano al giocatore, in attesa di essere giocate.

 Mano
- manoAttuale: ArrayList<Carta>
+ Mano() + getManoAttuale(): ArrayList<Carta> + aggiungiCartaAllaMano(carta:Carta):void + rimuoviCartaDallaMano(carta:Carta):void

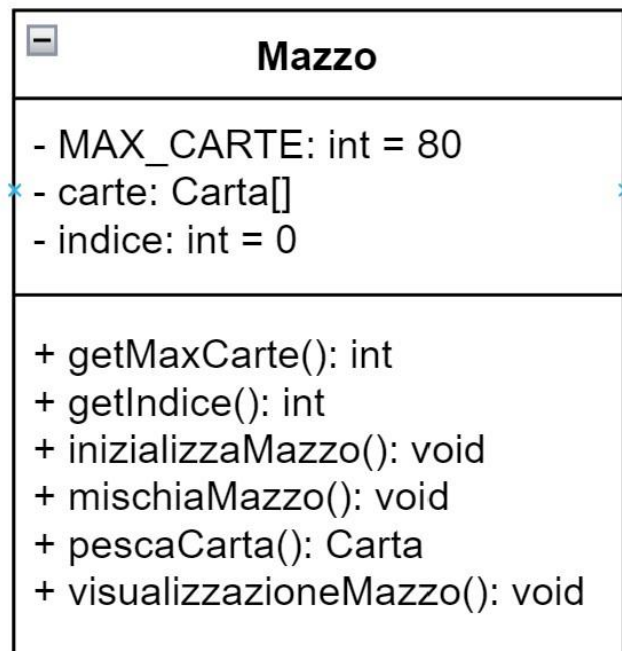
Classe Schieramento

Questa classe definisce l'insieme delle carte schierate dal giocatore sul tavolo di gioco.

 Schieramento
- carteSchierate: ArrayList<CartaServitore>
+ Schieramento() + getCarteSchierate(): ArrayList<CartaServitore> + schieraCarta(carta:CartaServitore):void + rimuoviCartaSchierata(carta:CartaServitore):void


Classe Mazzo

Questa classe definisce la struttura del mazzo in comune tra i due giocatori e dal quale vengono pescate le carte.



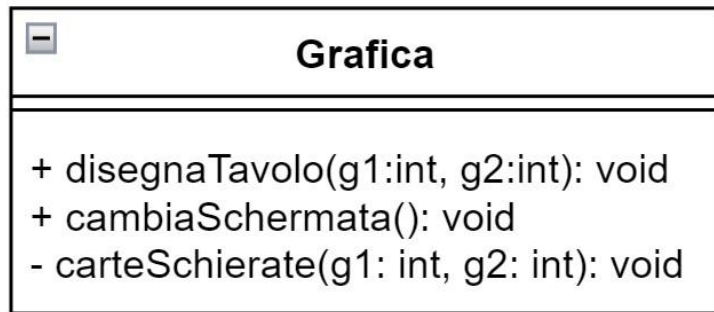
Classe Gioco

All'interno di questa classe sono definite fondamentalmente tutte le regole di gioco.

 Gioco
<ul style="list-style-type: none">- giocatore: Giocatore[]- carteAbilitate1: ArrayList<CartaServitore>- manaMaxAttuale: int- bestiaAttacco1: ArrayList<String>- saluteGialIncrementata: ArrayList<String>
<ul style="list-style-type: none">+ getGiocatore(num:int): Giocatore+ intro(): void+ inizializzaGioco(): void+ avviaTurno(): void- schieraCarte(num:int): void- utilizzaCarteSchierate(att:int, dif:int): void- verificaCartaDaUtilizzare(carta:CartaServitore, att:int, dif:int): void- verificaCartaGuerriero(carta:CartaServitore):void- azione(carta1:CartaServitore, carta2:CartaServitore, giocatore1:int, giocatore2:int): void- azione(carta: Carta, num: int): void- confrontoTipoCarte(cartaAtt: CartaServitore, cartaDif: CartaServitore): void- incrementaSalute(num: int): void- attacco(cartaUtilizzata: CartaServitore, att: int, dif: int): void+ fine(): void

Classe Grafica

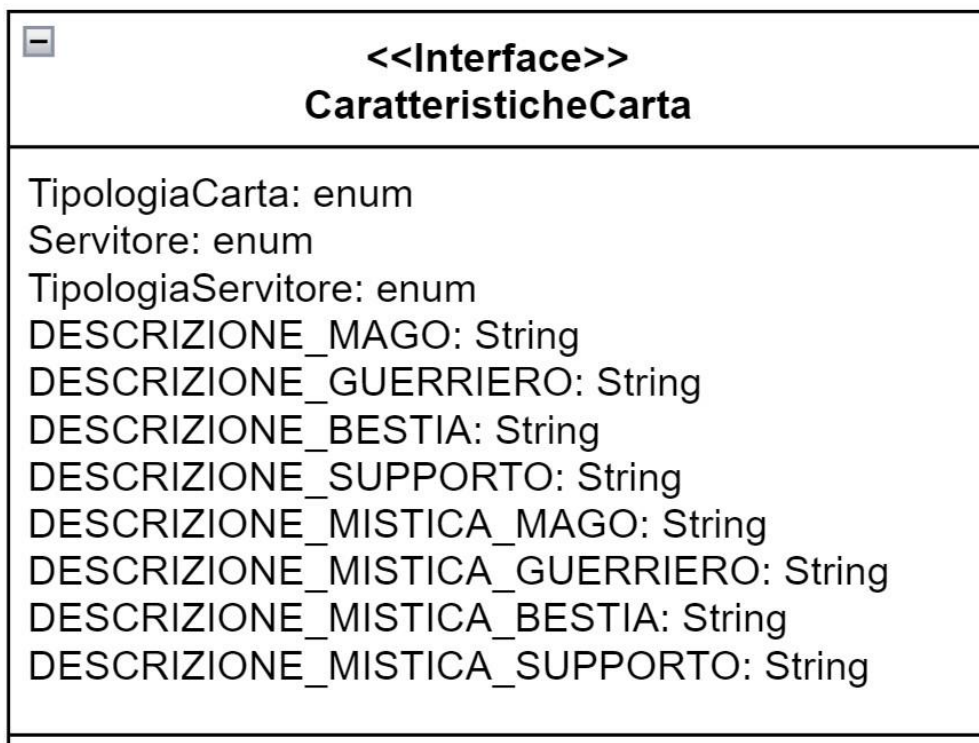
Questa classe si occupa di definire un output grafico, anche se abbastanza spartano, per far giocare tra di loro i due giocatori.



Interfacce

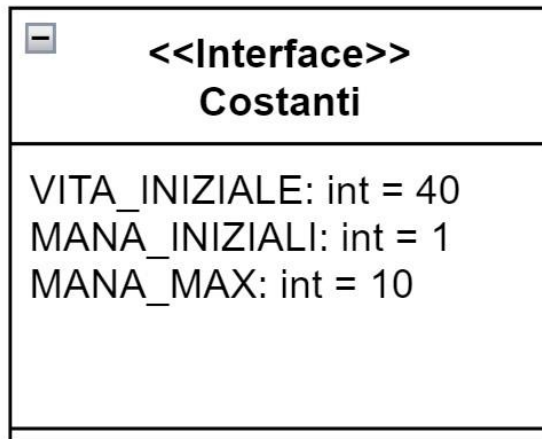
Interfaccia CaratteristicheCarta

Questa interfaccia è utilizzata per permettere a classi non correlate e sottoclassi non della stessa superclasse, di condividere le stesse informazioni sulle carte.



Interfaccia Costanti

In questa interfaccia sono definite le costanti utili per l'inizializzazione dell'oggetto Giocatore.



Class Diagram

