**Documentazione tecnica**

Javax.swing.JOptionPane = importa il pacchetto necessario per aprire una finestra di dialogo

* **Classe Perudos()**
  + **Variabili:**
    - **private** Giocatore[] giocatori;

è un array contenente il numero di giocatori. Questa variabile fa riferimento alla classe Giocatore().

* + - **int** giocatoreAttivo; // indice del giocatore in turno

indica il giocatore in turno. Questo è un int in quanto indica la posizione del giocatore all’interno dell’array.

* + - Scommessa scommessaAttiva;

Indica l’ultima scommessa emessa. La classe di riferimento è Scommessa().

* + - Scommessa scommessaPrecedente;

Indica la scommessa precedente a quella attiva. La classe di riferimento è Scommessa().

* + - String[] nomiGiocatori = { "Mario", "Giovanna", "Marco", "Giuseppe", "Francesca"};

Array di Stringhe contenente il nome dei giocatori.

* + **Metodi**
    - GetNumGiocatori()

Ritorna la lunghezza dell’array giocatori.

* + - GetNumDadiInGioco()

Calcola i dadi presenti nel gioco.

* + - StartGame()  
      Chiediamo all’utente con quanti opponenti vuole giocare.

L’array contenente il numero di giocatori massimo, chiamato possibilities contiene i numeri da 1 a 5. Successivamente facciamo visualizzare una finestra di dialogo Combo box in cui chiediamo al giocare con quanti avversari vuole giocare.

L’array di giocatori sarà uguale al numero di avversari + 1 (l’utente).

L’utente (o GiocatoreUmano()) si trova nella posizione 0 dell’array. Il nome dell’utente è : “Tu”

Tutti le altre posizioni dell’array sono occupate dal giocatore AI. Il nome verrà preso dall’array di stringhe numGiocatori. La posizione è [i-1] poiché il ciclo for parte da 1 ma la posizione occupata dal primo giocatore AI è 0.

All’inizio del gioco il giocatoreAttivo è nella posizione 0. Ad iniziare è quindi l’utente. Viene richiamato il metodo StartGiro().

* + - StartGiro()

Ad iniziare il giro è il giocatore che ha perso il dado (giocatori[i].InizioGiro()). Il primo giro viene iniziato dall’utente.

La scommessAttiva sarà data dal giocatore attivo. E quindi il giocatore collocato nella posizione “giocatoreAttivo” dell’array.

Chiamiamo i metodi PassaIlTurnoAlProssimo() e Loop().

* + - Loop()

Inizializziamo una variabile di tipo boolean giroFinito che ci indicherà quando il girò sarà concluso.

Il ciclo while sarà eseguito fin quando il giroFinito non sarà true.

All’interno del ciclo while poniamo scommessaPrecedente uguale alla scommessaAttiva. La scommessaAttiva sarà dichiarata dal giocatoreAttivo. Il giocatore può scommettere dubitando o rilanciando.

Se la scommessa Attiva è uguale a null ( significa che il giocatore ha dubitato) il loop sarà finito

Altrimenti richiamiamo il metodo PassaIlTurnoAlProssimo.

Adesso inizializziamo a 0 una variabile numDadi che incrementeremo successivamente. Inizializziamo anche una stringa tutti Dadi che ci mostrerà i dadi presenti nel gioco nel momento in cui un giocatore dubiterà.

All’interno del ciclo for contiamo quanti dadi ci sono della faccia scommessa. Salviamo quindi le stringhe di tutti i dadi dei giocatori. Se l’indice i si trova nella posizione 0 tuttiDadi stamperà “hai” poiche sta facendo riferimento al giocatore “Tu”. Ii caso contrario stamperà il nome del giocatore seguito da “ha”.

Dopo aver quindi mostrato i dadi in gioco verifichiamo chi ha perso la scommessa.

Nella condizione dell’if scriveremo infatti: se il numero di dadi è maggiore uguale alla scommessa fatta dal giocatore precedente, il giocatoreAttivo perde un dado ( o perde se ha solo 1 dado in gioco). Viene chiamata la funzione “Rimuovi DadoGioatore()”. Else (altrimenti, nel caso in cui la condizione dell’if non è soddisfatta) viene rimosso il dado del giocatore precedente che ha fatto la scommessa.

* + - RimuoviDadoGiocatore()

Se il giocatore precedente ha perso tutti i dadi viene eliminato. ( Se PerdiGiro() è false significa che non ci sono più dadi all’interno del bicchiere).

In caso contrario ricominciamo un nuovo giro. Se l’indice è 0 a perdere il dado è l’utente, altrimenti il giocatore successivo.

Ad iniziare il prossimo turno sarà il giocatore che ha perso il dado e la scommessa Attiva viene resettata.

* + - EliminaGiocatore()

Se l’utente non ha più dadi in gioco il gioco finisce con la sua sconfitta.

Se la lunghezza dell’array giocatori è minore di 3, il vincitore del gioco è l’utente.

In caso contrario il gioco continua e viene soltanto eliminato il giocatore AI che ha perso.

Creiamo quindi un nuovo array giocatoriCorrenti, minore di un elemento rispetto all’array giocatori. L’array contiene gli stessi elementi dei giocatori, meno il giocatore eliminato.

Il turno passa al giocatoreUmano che è di indice 0 e un nuovo giro (StartGIro()) viene cominciato.

* + - GetIndiceGiocatorePrecedente()

Restituisce il giocatore precendente a quello attivo. Questo ci serve per verificare chi ha perso il dado o il gioco. Se il giocatore si trova in una posizione maggiore di 0 (posizione dell’utente) il giocatore precedente sarà nella posizione precedente a quella del giocatoreAttivo (giocatoreAttivo -1).

Nel caso in cui invece la posizione del giocatore sia 0, il giocatore precedente è l’ultimo presente nell’array.

* + - PassaIlTurnoAlProssimo()

Passa il turno al giocatore successivo. Se il giocatoreAttivo non si trova nell’ultima posizione dell’array il giocatoreAttivo sarà incrementato di 1. In caso contrario il prossimo sarà nuovamente l’utente.

* **Classe Giocatore()**
  + **Variabili:**
    - **protected** Bicchiere bicchiere;

Questa variabile ci servirà per assegnare ad ogni giocatore un bicchiere

* + - **protected** String nome;

E’ una variabile di tipo String che indicherà il nome di ogni giocatore.

* + **Metodi**
    - GetNome()

Ritorna la variabile nome.

* + - SetNome()

Indica che la variabile nome a cui stiamo facendo riferimento è quella appartenente alla classe Giocatore().

* + - InizioGiro()

Ogni giocatore mescola i dadi all’interno del bicchiere.

* + - DecidiMossa()

Ritorna una variabile scommessa di classe Scommessa.

* + - Scommetti()

Ritorna null poiché questo metodo verrà implementato nelle classe sottostanti.

* + - Rilancia()

Ritorna una scommessa di classe Scommessa().

* + - ContaDadiConFaccia()

Conta i dadi con la stessa faccia presenti all’interno del bicchiere del giocatore.

* + - ScopriDadi()

Ritorna StringaValoreDadi(). Vengono quindi mostrati tutti i dadi dei giocatori avversari.

* + - GetNumDadi()

Ritorna il numero di dadi all’interno del bicchiere del giocatore.

* + - PerdiGiro()

Il giocatore perde un dado. Viene richiamato il metodo TogliDado() della classe bicchiere.

* **Classe GiocatoreUmano**
  + **Metodi:**
    - Scommetti()

Vengono mostrati al giocatore umano i giochi presenti all’interno del suo bicchiere.

CreaScommessa è null poiché non abbiamo ancora nulla da valutare, visto che l’utente sta facendo la prima mossa.

* + - CreaScommessa()

Qui il giocatore umano decide la sua scommessa.

Permettiamo al giocatore umano di fare solo scommesse valide.

Se c’è stata una scommessa precedente, allora non si può scegliere una faccia minore di quella precedente.

Inizializziamo quindi due variabili: numFacceValide con valore 6 e faccia Iniziale con valore iniziale 1.

Se la scommessa Precedente è diversa da null (c’è quindi stato qualcuno che ha scommesso prima di noi) la faccia Iniziale che potrà scegliere il giocatore partirà dalla faccia che ha scommesso il giocatore precedente.

Il numero di facce valide e' pari al numero di facce del dado (6), meno la faccia attuale della scommessaPrecedente, + 1 perché la faccia attuale è compresa tra le scommesse valide.

Nella finestra JOptionPane chiediamo quindi all’utente la faccia che desidera scegliere.

Non si può scommettere che ci siano più dadi di quanti ce ne sono effettivamente in gioco quindi giocatore potrà solo scegliere un massimo di dadi pari al numero di dadi in gioco.

Se l'utente rilancia ( c'era già una scommessa precedente),può o aumentare il numero di dadi della scommessa precedente, e mantenere la stessa faccia oppure aumentare la faccia di uno (se la faccia e' minore di 6).

* + - DecidiMossa()

Il giocatore umano può decidere di scommettere o dubitare. “n” sta ad indicare se il giocatore preme dubito (0) o rilancio (1).

Se il giocatore dubita allora la scommessa sarà null e si scopriranno tutti i dadi, altrimenti il giocatore potrà rilanciare.

* + - Rilancia()

Mostra all’utente i propri dadi e il giocatore umano decide la scommessa.

* **Classe GiocatoreAI()**
  + **Variabili** (non sono altro che le personalità dei vari giocatori):
    - **private** **boolean** bluffatore = **false**;

La variabile booleana restituirà true se l'AI bluffa, ovvero scommette dadi che non ha nel proprio bicchiere, false altrimenti.

* + - **private** **float** dubbioso = 0.0f;

Probabilita' di quanto sia dubbiosa l'A.I. (valore da 0-non dubbioso, a 1- dubbioso)

* + - **private** **float** azzardoso = 0.0f;

Percentuale di quanto sia azzardosa l'A.I. 0= cauta, 1= azzardosa

* + **Metodi:**
    - Scommetti()

Inizializziamo due variabili faccia Frequente e numDadi necessari a far scommettere l’A.I. in modo intelligente.

Controlliamo quanto è azzardosa la nostra A.I.. Se azzardoso è minore di 0.3 (quindi è cauta) farà una scommessa bassa. Scegliendo la faccia più frequente presente all’interno del proprio bicchiere e dichiarandone una quantità minore rispetto a quella dei propri dadi con tale faccia. Se non ha nessun dado con tale faccia affermerà che la quantità di dadi è 1.

In caso in cui la variabile bluffatore sia true, aumenterà di 1 il numero di dadi presente. Se la variabile azzardoso è maggiore di 0.7 (molto azzardoso) farà lo stesso.

* + - DecidiMossa()

Per prima cosa controlliamo se AI ha dadi uguali alla scommessa.faccia (la scommessa fatta precedentemente).

Controlla la scommessa e decide se dubitare o meno.

La scelta di dubitare deriva dalla percentuale di dubbiosità dell’AI.

* + - Rilancia()

Controlliamo se AI ha dadi uguali alla scommessa.faccia. Se la nostra A.i. è azzardosa, preferirà rilanciare la faccia più vicina alla faccia più frequente.

* **Classe Bicchiere()**
  + **Variabili:**
    - private static int *NUM\_FACCE* = 6;

E’ una variabile statica che indica il numero Massimo di facce di tutti i dadi. Questa variabile non verrà mai modificata.

* + - private static int *NUM\_DADI\_INIZIALI* = 5;

E’ una variabile statica che indica il numero di dadi iniziali di ogni giocatore. Questa variabile verrà eventualmente decrementata se il giocatore perderà il turno.

* + - private Dado dadi[];

E’ un array di dadi che fa riferimento alla classe Dado.

* + - private int contatoreFacce[];

E’ un array di int che conta quante facce di uno stesso tipo ci sono all’interno del gioco. Serve per verificare chi dei giocatori deve perdere il dado perché ha dubitato o affermato una quantità maggiore rispetto a quella realmente presente.

* + - private int facciaFrequente;

Questa variabile serve al giocatore AI per scommettere o rilanciare la quantità di dadi più congeniale alla sua vittoria.

* + **Metodi:**

## GetNumDadi()

Ritorna la lunghezza dell’array contenente il numero di dadi.

* + - GetValoreDadi()

Ritorna un array di int chiamato valori Dadi che mostra le facce dei dadi.

* + - StringaValoreDadi()

Ritorna una Stringa contenente le facce dei dadi. Tale stringa verrà poi mostrata nel momento in cui un giocatore dubiterà.

* + - TogliDado()

restituisce true se è possibile togliere un dado, false se non ci sono più dadi nel bicchiere.

* + - MescolaDadi()

Mescola i dadi all’interno di ogni bicchiere con la funzione mescola

Richiamata dalla classe Dadi.

* + - ContaFacce()

mantiene la conta di quante facce dello stesso tipo ci sono nei dadi del bicchiere. ContatoreFacce[i] è uguale al numero di dadi nel bicchiere

la cui faccia è i+1 ( perchè l'array parte da 0).

Successivamente con facciaFrequente troviamo il numero che si ripete più volte all’interno del bicchiere.

* + - GetFacciaFrequente()

Aggiorniamo il numero di faccia Frequente.

* + - GetNumDadiConFaccia()

Conta il numero di dadi con una stessa faccia presenti all’interno del bicchiere.

* **Classe Dado()**

L’unica variabile che ci interessa è faccia, indicante la faccia del dado.

Il metodo Mescola() prende un numero, da 1 a 6, random e l’assegna alla variabile faccia. In questo modo creeremo facce random nei 5 dadi di ogni giocatore.

Il metodo GetFaccia() ritorna la faccia del dado.

* **Classe Scommessa()**

Prendiamo in considerazione 2 variabili che inidicano la faccia e il numero di dadi.

Poiché sono variabili private abbiamo bisogno di metodi pubblici che possano essere richiamati da altre classi. Creiamo quindi GetFaccia() e GetNumDadi().

Con il SetFaccia() e SetNumDadi() riferiamo la variabile del metodo a quella della classe.

* **Classe Main()**

L’unica cosa che viene fatta è inizializzare il gioco chiamando il metodo StartGame dalla classe Perudos. Per fare ciò creiamo un oggetto statico perudo di classe Perudo() in modo tale che le altre classi lo possano referenziare.