

CLASES DEL PROYECTO BLACKJACK (21)

Carta.java

La clase Carta es la encargada de definir cada una de las cartas que van a definir la baraja.

Con un entero 'palo' y otro entero 'valor' somos capaces de definir una carta de forma inequívoca.

También definimos una serie de constantes para definir los posibles palos y los posibles valores de las cartas que nos vamos a encontrar. Como se puede observar, para las cartas comprendidas entre el 2 y el 10 no es necesario definir ninguna constante, ya que el propio número de la carta será su valor.

La clase define otra serie de métodos auxiliares utilizados para obtener la propiedad de cada carta.

Podemos obtenerla tanto en formato entero, que es como se ha definido como propiedad de la clase; o también tenemos otra pareja de métodos que nos devuelven las propiedades de la carta como si fuera un String.

Estos métodos son getPalo(), getValor(), getPaloString() y getValorString().

Otro método toString() nos devuelve un String con la descripción del palo, es decir, "As de Diamantes" o "4 de Picas", lo cual nos será útil para decirle al jugador las cartas en cuestión.

CLASES DEL PROYECTO BLACKJACK (21)

Baraja.java

La clase Baraja define simplemente una baraja de cartas, que lo simularemos con un array de objetos Carta. Está formada únicamente por 4 métodos, uno de ellos el constructor, encargado de generar toda la baraja de 52 cartas. Con un par de bucles for vamos recorriendo los posibles palos y los posibles valores hasta generar la baraja completa. Definimos otra propiedad 'restantes' para saber cuantas cartas nos quedan en la baraja. El método barajar() se encarga de ordenar la baraja.

Para ello utilizaremos el método random de la clase Math y una variable auxiliar.(investigar y estudiar EN INTERNET PARA UN MEJOR ENTENDIMIENTO)

Por último, el método restantes() nos dice cuantas cartas quedan en la baraja y el método robar() coge una carta de la baraja, la primera del mazo.

Dado que en el constructor no barajamos la baraja, lo haremos en el método de robar en caso de que la baraja esté completa.

También nos definimos un array auxiliar llamado vistas, que utilizará la banca para decidir si coge nueva carta o no. Este array tiene tamaño 13 y en cada una de las posiciones tiene el número de cartas, cuyo valor es dicha posición, que se encuentran descubiertas en la mesa o las posee la banca.



CLASES DEL PROYECTO BLACKJACK (21)

Mano.java

Utilizaremos la clase Mano para indicar el número de cartas que va a tener un jugador y la banca en la mano.

Para ello utilizaremos un Vector de objetos Carta.

No utilizamos un array como en la clase Baraja, porque en este caso no sabemos cual va a ser el número de cartas que va a tener cada participante del juego en su mano.

En la clase Mano también tendremos la información sobre el dinero apostado por dicho jugador en esta ronda del juego.

Aquí definiremos todas las alternativas que puede tomar el jugador durante la partida cuando tiene un conjunto de cartas en su mano. Vamos a explicar brevemente cada una de ellas.

Para empezar, los métodos cogerCarta() y dejarCarta() realizan las acciones que sus nombres indican. En el caso de dejarCarta() podemos soltar una carta determinada o la que se encuentre en una posición indicada. El método soltar() suelta todas las cartas que tiene el jugador en la mano.

El método contar() cuenta las cartas que tiene el jugador en su mano y el método obtenerCarta() devuelve la carta situada en una posición específica.

Por último, el método más importante de esta clase es el getBlackjackValor(). Este método se encarga de contar la puntuación de las cartas que tiene el jugador en su mano. Para eso se basará en que las figuras tienen un valor de 10 puntos, las cartas entre el 2 y el 10 tienen el valor especificado en su número y el as tendrá un valor u otro en función de la jugada. Como ya se ha explicado antes, el as valdrá 11 puntos en caso de que la suma con



CLASES DEL PROYECTO BLACKJACK (21)

el resto de cartas de la mano sea menor o igual que 21.
En caso contrario valdrá 1 punto.



CLASES DEL PROYECTO BLACKJACK (21)

Blackjack.java

La clase Blackjack es la clase principal del programa, ya que en ella se encuentra el método main() que va a llevar el desarrollo del juego.

Lo primero que se hace al entrar en el método main() es presentarle al usuario el juego y sus autores, y a continuación preguntar cual va a ser la cantidad inicial de dinero en euros de la que va a disponer cada jugador y el número de jugadores que van a participar en el juego.

Una vez hecho esto comienza el juego y el programa entra en un bucle infinito del que solo se saldrá si todos los jugadores desean interrumpir la partida o si se quedan sin dinero.

Inmediatamente después se le preguntará a cada jugador cuanto dinero desea jugar y se le presentará un mensaje de error en caso de que intente jugar con más dinero del que tiene o en caso de que introduzca una apuesta negativa.

Una vez comprobado que la apuesta es correcta y que todavía quedan jugadores en la partida (no se han arruinado todos o han abandonado) podemos pasar al método jugar(), que es el que va a llevar la partida.

Lo primero que haremos en el método jugar() será repartir 2 cartas a cada jugador y otras 2 a la banca. Antes de imprimir por pantalla las cartas del jugador y la carta descubierta de la banca se comprobará si alguno de los jugadores o la banca suma 21 puntos (Blackjack) en cuyo caso se termina el juego y se vuelve a realizar una nueva apuesta.

En caso de que no haya Blackjack se muestran las cartas a cada jugador, por orden, y este decide si se planta o si sigue jugando. El jugador introduce un



CLASES DEL PROYECTO BLACKJACK (21)

carácter 'C' para pedir nueva carta o un 'P' para plantarse. En caso de que introduzca un carácter incorrecto se le mostrará un mensaje de error.

Si el jugador decide plantarse, el turno pasará a la siguiente y si decide coger una nueva carta se le añadirá a su mano y, si no supera los 21 puntos. En caso de superar los 21 puntos el jugador habrá perdido.

Este proceso se irá repitiendo para todos los jugadores involucrados en la partida.

Una vez que todos los jugadores hayan jugado, es el turno de la banca. En los casinos, la banca tiene la norma de en cuanto supere los 16 puntos debe plantarse, pero nosotros hemos hecho un algoritmo más elaborado para dotarle de más inteligencia.

Cuando sea el turno de la banca, esta deberá decidir si roba nueva carta o no. Para decidirse comparará 2 posibles situaciones.

La primera de ellas será calcular el beneficio que obtendría en la situación actual, teniendo en cuenta su puntuación, la de los demás y los jugadores que ya se han pasado de 21 puntos.

Para calcular lo segundo nos basaremos en la probabilidad de obtener una carta u otra al robar del mazo. Para ello, calcularemos la probabilidad de obtener la carta x y el beneficio que obtendríamos con dicha carta. Sumando todas las posibilidades obtendríamos un beneficio hipotético.

Beneficio hipotético = $\sum \text{probabilidad}(\text{carta } i) \cdot \text{Beneficio}(\text{carta } i)$

Por tanto, comparando el beneficio hipotético con el beneficio actual podemos decidir si cogemos una nueva



CLASES DEL PROYECTO BLACKJACK (21)

carta o no.

Una vez hecho esto deberemos ver quien ha ganado. Todos compiten contra la banca, luego esta comparación deberemos hacerla entre cada uno de los jugadores y la banca.

Durante el programa, siempre que algún jugador gana o pierde asignamos un valor a un entero dentro de un array llamado 'resultados' y salimos del bucle. El valor de dicho entero determina quien ha ganado y en qué condiciones.

En la siguiente tabla presentamos los posibles valores de dicho entero y lo que significa cada uno:

Valor Significado

- 0 Situación inicial
- 1 He ganado
- 1 He perdido
- 2 He logrado Blackjack

