

**Card**

- Suit suit (enum);  
- Value value (enum);

+ card constructor (suit s value v;  
+ getter  
+suit()  
+value()  
+toString()

**Suit**

enum Suit  
(CLUBS, HEARTS, SPADES, DIAMONDS)

**Value**

enum Value  
(ACE, 2,3,4,5,6,7,8,9,10 JACK , KING, QUEEN

**Deck**

- Set<card> deck;  
- int numberOfCards;

+ getter  
+numberOfCards  
+ deck constructor()  
+ contains ()  
+ draw {return card}

**Hand**

- ArrayList<card>  
- int numberOfCards;

+ add()  
+ getter  
+ getNumberOfCards  
+ Hand constructor()  
+ remove() {return card}

**Player**

- double balance;  
- String name;  
- Hand hand;

+ getters  
+ reduceMoney (double amount);  
+ increaseMoney (double amount);  
{return boolean}

**Dice**

- int Value

+ getters  
+ rollDice(int numberOfDice);

**Card Dealer**

- Hand hand  
- Deck deck

+ getters  
+ Dealer constructor ()  
+ deal (int noumberOfCards);

**Dealer****Dice Dealer**

- Dice dice

+ getters  
+ Dealer constructor ()  
+ deal int rollDice();

**Casino**

- ArrayList<Player>  
- Game[]

+ main method

**User Interface**

- Scanner scanner  
Class used to store input and output messages

**Bettable**

- giveMoney (player);  
- takeMoney (player);

Game (abstract)
implements Bettable + playGame();

Card Game (abstract)
abstract compareCard( Hand, Hand)

BlackJack Handler
Player player; + hit(); +clearHand();

BlackJack (Games themselves implements Bettable not handler)
implements Bettable - Handler handler; - Dealer dealer; HashMap<Value, Integer> (ACE = 1, ACE = 11)
+ BlackJack constructor() + CompareCards (Hand h player, hand dealer) + playGame(); + computeHandValue()

H/L Card
extends CardGame HashMap<Value, Integer> - Handler handler - Dealer dealer
+ H/L constructor + CompareCards( Hand h, Hand h2) + playGame()

H/L Handler Card
implements Bettable Player player
+ giveMonery() + takeMoney() + addCard() + clearHand()

Card Handler (abstract)
Hand hand; + addCard(); + clearHand();

Handler
Player player (protected)