CardTable <<Java Class>> <<Java Class>> <<Java Class>> <<Java Class>> Assign6Phase3 (default package) CardGameFrameWork MAX_CARDS_PER_HAND :int MAX_PLAYERS :int MAX PLAYERS: int int counter boolean runTimer PAUSE= 100 iconCards: icon[][] Assig6Phase3() iconBack: icon iconsLoaded: boolean numPlayers:int numPacks: int numJokersPerPack:int numCardsPerHand:int numPlayers:int pnlComputerHand: JPane main(String[]):void Assig6Phase3() GUICard() numUnusedCardsPerPack : int numCardsPerHand : int Clock() formatToTime():string timerEvent():ActionListiner buttonEvent():ActionListiner loadCardIcons():void turnIntIntoCardValue(int):String pnlHumanHand: JPane pnlPlayArea: JPane main(String[]):void valueAsInt (Card):int turnIntIntoCardSuit (int):Suit CardGameFrameWork (int, int, Card int, int) CardGameFrameWork () suitAsInt(Card):int getHand(int):Hand getCardFromDeck ():Card getNumCardsRemainingInDeck ():int newGame():void getlcon(Card):lcon getBackCardlcon():lcon <<Java Class>> HighCardListiner deal():boolean sortHands():void HighCardListiner() actionPerformed (ActionEvent):void playCard (int): Card takeCard (int): boolean <<Extends Thread>> TimerClass <<Java Class>> Hand <<Java Class>> run() doNothing() Deck MAX CARDS:int MAX CARDS: int numCards:int isExecuted: boolean topCard: int Hand() numPacks:int resetHand():void takeCard(Card):boolean Deck(int) playCard (int): Card Deck() playCard():Card toString():String getNumCards ():int init(int):void getNumCards ():int inspectCard (int): Card sort():void shuffle():void dealCard():Card getTopCard():Card addCard(Card):boolean removeCard (Card):boolean sort():void inspectCard (int): Card <<Java Class>> Card cardValue: char[] valueRanks: char[] value: char errorFlag: boolean card*char, Suit) Card() toString():String set(char, Suit):boolean getValue():char getSuit():Suit getErrorFlag():boolean equals(Card):boolean arraySort(Card[], int):void <<Java Enumeration>> Suit clubs: Suit diamonds: Suit hearts: Suit spades: Suit

Suit()