Gordon Tenev

Activity2

1. A deck is a list of cards.
2. Six
3. Ranks: {“Two”, ”Three”, ”Four”, ”Five”, ”Six”, ”Seven”, ”Eight”, ”Nine”, ”Ten”, ”Ace”, ”Jack”, ”Queen”, ”King”}  
   Suits: {“Hearts”, “Spades”, “Clubs”, “Diamonds”}  
   pointValues: {2,3,4,5,6,7,8,9,10,11}
4. Yes

Activity3

1. 3,2,1,0

Activity6

1. 5♠,6♣  
   6♣, 5♣
2. No, you can have 2 matching numbers it’s just you will lose after you do that
3. Sorta but not really. In certain cases it is better to pick one pair over the other, i.e. if you have two cards of the same number that match another card and a card of a different number that matches a different card, go with the card with two numbers since the net values discarded will be less. But really it’s just luck, what a stinky game

Activity7

1. A deck of cards, cards, a counting system to see how many times I win
2. Create a new deck of cards  
   REPEAT UNTIL LOSE OR DECK EMPTY{  
   Let the player pick the card  
   Check to see if it is good  
    If not, pick another pair  
    If it is, fill the empty spot with new cards  
   Check to see if there are any more matches  
    If not, exit loop  
   }
3. I think so
4. 1. In the constructor and when a new game is made after the shuffle. Also anotherPlayIsPossible().
   2. isLegal(), any other methods can implement isLegal() instead of checking for paris or JQKs.
   3. {0,1,2,3,4,5}
   4. for(Integer : cIndexes) { System.out.print(Integer + " ");
   5. anotherPlayIsPossible(), to see how many cards are actually not taken.

Activity8

1. They all utilize the same type of boards, it’s just you’re trying to get things to add to 10s, 11s, 13s, etc.
2. Its constructor method sets all its cards to equal what the elevensboard requests it to be:  
    public ElevensBoard() {  
    super(BOARD\_SIZE, RANKS, SUITS, POINT\_VALUES);  
    }
3. Yes, as they cover if there is a legal move and if another play is possible, both dependent on whether your goal is to add up to a 10, 11 or 13.

Activity9

1. Because even though the size varies per board, all that method does is output the size. Thus, since the size was already initialized in the beginning of the program, we simply need to return the value; thus it is the same protocol.
2. Again, same protocol. Simply remove the cards that are selected. You check to see if it is valid or legal to do that. That method is the one that is abstract. The actual removing is the same throughout.
3. You can call all methods polymorphically but using an interface isn’t as practical. Since some of the methods like the ones stated above will be the same for any 10s, 11s, 13s, etc. game, it would be more practical to implement them all in an abstract class rather than an interface.

Activity11

1. Around 0-10% success rate oof
2. Around 5-15% success rate MEGA oof
3. I did 100,000 and got consistently about a 10.5%-10.8% success rate.