Rules of the game

A game of Solitaire consists of

- the main deck, which is a pile of cards with the top card turned over
- seven tableau piles of cards
- four foundation piles of cards

A tableau pile consists of 0 or more hidden cards at the bottom of the pile. If a tableau pile is non-empty, then it has at least one exposed card on the top of the pile. A tableau pile may have more than one exposed card. Note that a tableau pile never contains a hidden card placed on top of an exposed card.

At the beginning of the game, the first tableau pile has one card, the second has two cards, etc.

A foundation pile contains cards of the same suit, arranged in order from Ace at the bottom of the pile to King at the top of the pile. (In Klondike, Aces are low.) There are four foundation piles, one for each suit.

On each turn, a player may either

- draw another card from the main deck
- move one or more cards

Drawing a card from the main deck means removing the current top card and placing it in a waste pile. The new top card on the main deck is then exposed. If the main deck is empty, then all of the cards are transferred from the waste pile back to the main deck. (Following the transfer of cards from the waste pile back to the main deck, they should appear in the order in which they originally occurred.)

Moving a card transfers one or more cards from either the main deck or a tableau pile to a tableau pile or a foundation pile. Moves must be done following the rules of the game, which are as follows:

- Only the top card may be removed from the main deck.
- Only exposed cards may be moved. (Hidden cards may not be moved.)
- The cards moved are placed on top of whatever pile they are moved to.
- When multiple cards are moved (from a tableau pile), the cards cannot be removed from the "middle" of the pile. Instead, a chosen card and all cards on top of it must be moved.

- The colors of the cards in a tableau pile must alternate red and black. Clubs and Spades are black suits, and Diamonds and Hearts are red suits.
- When a card or cards are moved onto an empty tableau pile, the new bottom card must be a King.

Following a move, the top card of the pile the card or cards were moved from is exposed (if the pile is not empty.)

Program requirements

Your program should shuffle the main deck, deal cards into the tableau piles, and then allow the user to play the game by repeatedly drawing the next card from the main deck, or moving exposed cards. Before each turn, the program should print a representation of the game state showing

- how many cards are in the main deck,
- which card is on top of the main deck
- the top card in each foundation pile
- all hidden and exposed cards in each tableau pile (print "xx" for hidden cards)

User input must be validated. If the user enters invalid input, the game state should be printed again and the user repromoted to enter valid input.

All of your classes and methods must be well documented.

Your methods must contain appropriate comments. I expect to see comments describing important sections of code: for example, to determine when a move is legal, to move cards, the check to see if the user has won the game, etc.

All comments should be well-written and professional.

Milestones

There are two milestones.

Milestone 1: The program must create the initial tableau piles, and allow the player to either draw a card from the main deck or move a card. If the player chooses a move, the program must check the validity of the move. However, the program does not need to carry out the draw or move: that is the task in Milestone 2.

Milestone 2: Allow the player to play a complete game. When a legal move is chosen, the move should be carried out by moving cards between piles as appropriate according to the rules of the game. The player should detect when the player has won and print a message as appropriate.

String methods

You can add a **String** members w.r.t to objects of classes where you want to provide a way to mention strings for printing

CardLocation

An instance of the **CardLocation** class represents one of the cards the player can choose to move. Its data consists of the index of one of the piles, and the index of a card within the chosen pile.

Game class

Write a class called **Game** representing the current state of the game. This class can have methods to determine which moves are legal, and to carry out those moves.

Here are some of the methods common in this **Game** class:

- public Deck getMainDeck()
- public Deck getPile(int index)
- public boolean isFinished()
- public boolean isValidTakeLocation(CardLocation takeLocation)
- public boolean isValidMove(CardLocation takeLocation, int placePileIndex)
- public void moveCards(CardLocation takeLocation, int placePileIndex)
- public void drawNextCard()

The most interesting methods are **isValidTakeLocation** and **isValidMove**. **isValidTakeLocation** determines if the card selected by the user can be legally moved. **isValidMove** determines if it is legal to move the card selected by the user to a particular pile.

The **moveCards** method moves the cards selected by the user to a particular pile.

Deck class

You can use instances of the **Deck** class to represent the piles. Since the piles are numbered (0 = main deck, 1-7 = tableau piles, 8-11 = foundation piles), an array of references to **Deck** objects is a good way to store the **Deck** objects representing the piles.

You can also use a **Deck** object to represent the waste pile.

If you use the **Deck** class to implement your piles, you will want to move the code in the constructor which adds the 52 cards to a separate method. The **Deck** representing the main deck should start out with 52 (shuffled) cards. You can use the **drawCard** method on the main deck to deal cards to the seven tableau piles.

Here are some of the methods common to **Deck** class:

- public void add(Card card)
- public Card getTopCard()
- public void setExposeIndex(int index)
- public int getExposeIndex()
- public int indexOfTopCard()
- public ArrayList<Card> removeCards(int index)
- public void placeCards(ArrayList<Card> cardsToPlace)

The "expose index" of a deck is the index of the first exposed card in the deck. For example, if a deck has 6 cards and the expose index is 3, then the cards with indices 3, 4, and 5 are exposed.

The **removeCards** and **placeCards** methods are used to move cards from one pile to another. **removeCards** removes all cards starting at the one whose index is given. **placeCards** places a sequence of cards on top of a deck.

Grading

Milestone 1:

- Initializing and printing the initial state: 50
- Allow player to choose a move or draw: 10
- If a move is chosen, check whether it is legal: 30
- Good coding style: 5
- Good comments: 5

Milestone 2:

- Player can move cards (for legal moves): 30
- Support drawing next card from main deck (adding current top card to waste pile): 20
- Waste pile is recycled when the main deck becomes empty: 10
- Game loop (player can make progress my making a series of moves): 25
- Checking end of game condition: 5
- Good coding style: 5
- Good comments: 5

Credit may be deducted for poor programming style (inconsistent indentation, poorly-chosen variable/method names, failure to make fields private, unnecessary use of static fields, etc.)