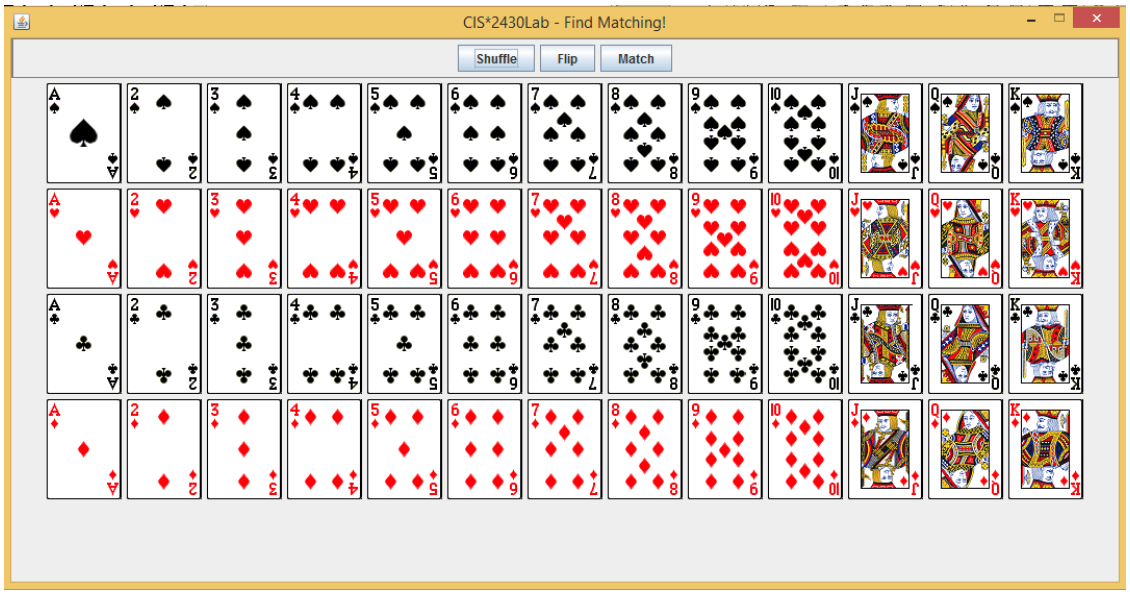
CIS2430 - LAB3 Card Matching Game



Description

In this Lab, the TAs will work with you to make a simple card game, called "Find Matches". Your responsibility is to provide the fundamental functionalities to support the game, and the TAs will take care of the GUI programming. Because it is a collaborative effort, all the implementations of classes, public methods and variables MUST comply with the following specification.

Specification

you are required to provide two classes: Card and Deck.

Card: Card class provides variables and methods to define and access each playing card's properties including suit, rank, front/back image, whether a card is faced-up or faced-down, etc.

Public methods & variables	description
enum Suit {**SPADES, HEARTS, CLUBS, DIAMONDS** }	Enum type that correspond to 4 suits of playing cards
Card(**Suit** suit, **int** rank)	Create a card with the given suit and rank where the rank of a card is defined as an integer for 'A'-1, '2'-2, ..., 'J'-11, 'Q'-12, and 'K'-13.
int getRank()	Return the rank of a card
Suit getSuit()	Return the suit of a card
void setFaceup(**boolean** faceup)	Flip a card to face-up or face-down
boolean getFaceup()	Return whether a card is face-up or face-down
Object getImage()	Return the attached image
void setImage(**Object** image)	Attach an image object to the card
static void setBackImage(**Object** image)	attach an back image to the entire Card class

static Object getBackImage()	Get the back image attached to the entire Card class
String toString()	Convert a card object into a string in the format: suit.name()+rank, e.g., "SPADE13", "CLUB2"
boolean equals(**Card** card)	Return true only if two cards have the same rank.

Deck: Deck class stores the entire pack of 52 cards (except the two 'Jokers') in an ArrayList, providing the following variables and methods

Public methods & variables	description
Deck()	create and initialize the entire pack of 52 cards in this constructor. How you order the cards in the ArrayList does not matter.
ArrayList<Card> getCards()	return the entire pack of cards in an ArrayList object
void **shuffle**()	Re-order all the cards in the ArrayList randomly, i.e., the same position in the ArrayList may contain different cards before and after calling shuffle().

Implementation

You are recommended to download the GUI code (which is ready in Neatbean project format) from Moodle and start programming from there. To generate class files for Card and Deck, open the project using Netbeans and choose [new file] ->[Java]->[Java class] .

Testing

Once all the specs are met, you shall be able to compile, run and test your implementation. If the main window shows up with all cards loaded properly, congratulations, you have the most parts done correctly. There are a few buttons to verify the correctness of your implementation.

'Flip' button (or mouse-click on each card) to inspect setFaceup(), getFaceup(), setBackImage(), getBackImage() methods of Card class

'Shuffle' button to inspect shuffle() method in Deck class.

'Match' button to start a simple game to inspect equals() method of Card class. The card with the same rank as the prompt card is a match.

Note: in order to load card images correctly, you should not change the image folder and also make sure you have implemented toString() method of Card class in compliance with the spec.