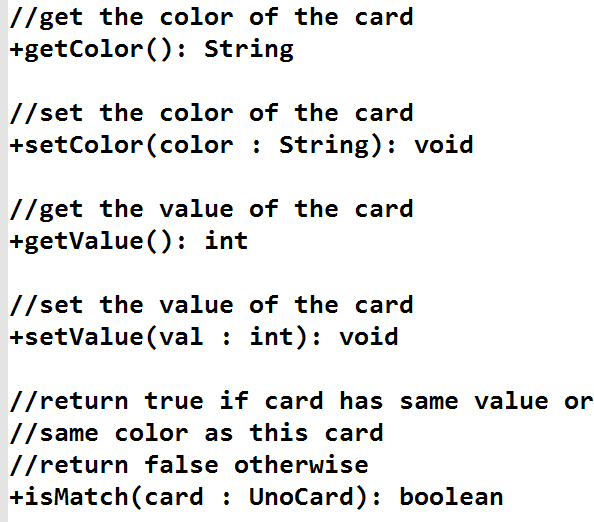
ITP277: Milestone 2

The goal of this milestone is to identify the objects needed for the UNO card game and the actions that those objects perform, rather than how the objects are actually represented. For example, a standard UNO card must be able to report both its color and value, and, when given another card, tell whether or not it is a "match." When specifying operations, think *simplicity*. "It's better to have **a few, simple operations** that can be combined in powerful ways, rather than lots of complex operations." (MIT, n.d.)

1. Activity Directions: Prepare a document that includes an ADT (abstract data type) for each object needed in your program. Create a LOOM video in which you explain your class design. Discuss your selection of properties and methods for each object.



Partial ADT for UnoCard class

1. Submit your document and a link to your video in brightspace. Ask your instructor for specific submissions details.

public class Deck {  
 protected Card topCard = new Card();  
  
 protected int cardsInDeck;  
  
 public Deck(Card topCard, int cardsInDeck) {  
 this.topCard = topCard;  
 this.cardsInDeck = cardsInDeck;  
 }  
  
 public Deck() {  
 }  
  
 public Card getTopCard() {  
 return topCard;  
 }  
  
 public int getCardsInDeck() {  
 return cardsInDeck;  
 }  
  
 public void setCardsInDeck(int cardsInDeck) {  
 this.cardsInDeck = cardsInDeck;  
 }  
}

public class Stack {  
 protected Card topCard = new Card();  
  
 public Stack(Card topCard) {  
 this.topCard = topCard;  
 }  
  
 public Stack() {  
 }  
  
 public Card getTopCard() {  
 return topCard;  
 }  
  
 public void setTopCard(Card topCard) {  
 this.topCard = topCard;  
 }  
}

public class Player {  
  
 protected int cardsInHand;  
  
 public Player(int cardsInHand) {  
 this.cardsInHand = cardsInHand;  
 }  
  
 public Player() {  
 }  
  
 public int getCardsInHand() {  
 return cardsInHand;  
 }  
  
 public void setCardsInHand(int cardsInHand) {  
 this.cardsInHand = cardsInHand;  
 }  
  
 public void drawCard(Deck d)  
 {  
 this.setCardsInHand(this.getCardsInHand()+1);  
 d.setCardsInDeck(d.getCardsInDeck()-1);  
 }  
  
 public void playCard(Stack s,Card c)  
 {  
 this.setCardsInHand(this.getCardsInHand()-1);  
 s.setTopCard(c);  
 }  
}

public class Card {  
  
 protected int value;  
 protected String color;  
 protected String action;  
  
 public Card(int value, String color) {  
 this.value = value;  
 this.color = color;  
 }  
  
 public Card() {  
 }  
  
 public int getValue() {  
 return value;  
 }  
  
 public void setValue(int value) {  
 this.value = value;  
 }  
  
 public String getColor() {  
 return color;  
 }  
  
 public void setColor(String color) {  
 this.color = color;  
 }  
  
 public String getAction() {  
 return action;  
 }  
  
 public void setAction(String action) {  
 this.action = action;  
 }  
  
 public boolean isMatch(Card c1, Card c2)  
 {  
 if(c1.getValue()==c2.getValue() || c1.getColor()==c2.getColor())  
 return true;  
 else  
 return false;  
 }  
}