IIP

Test Unit 5 - Possible solution

Year 2015-2016

Name:

1. Given the `Card` class with the following structure:

```
public class Card {
  private int value, color;
  public static final int OROS=0, COPAS=1, ESPADAS=2, BASTOS=3;

  public Card(int v, int c) {
    if (v>0 && v<13 && (c==OROS || c==COPAS || c==ESPADAS || c==BASTOS)) { value=v; color=c; }
  }
  ...
}
```

   Implement the following methods:

   a) A overriden version of the `toString` method that returns a representation of the card in the format "`VALUE of COLOR`", where `VALUE` is the value in number (1, 2, ..., 12) and `COLOR` must be "oros", "copas", "espadas", or "bastos" according the value. **You must employ the constant attributes when comparisons on color are done.**

   b) A `winner` method that returns a `boolean` value when the current card (that in `this`) wins to a card given as parameter (that it is supposed to be different in all cases); another parameter will be the dominant color. A card wins another card when:

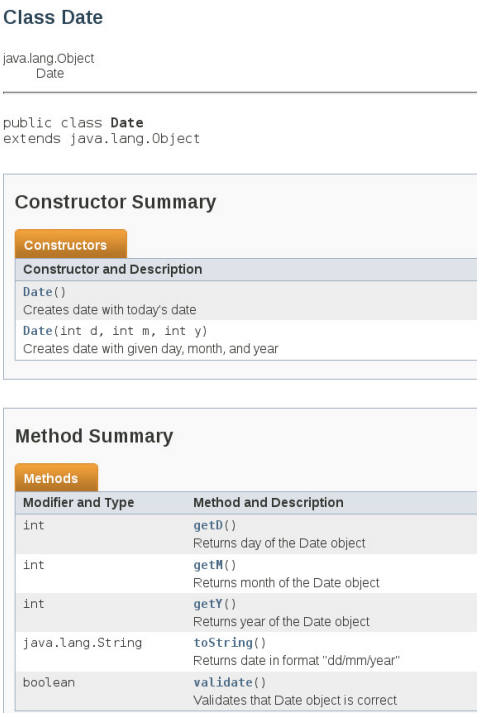
   - It is of the dominant color and the other card is of a different color
   - It has the same color than the other and the value is more powerful

   The most powerful value is 1, the second most powerful is 3, and the rest are in numerical order (i.e., 12 more powerful than 11, 11 more powerful than 10, etc.).

```
public String toString() {
  String res=value+" of ";
  switch(color) {
    case OROS: res+="oros"; break;
    case COPAS: res+="copas"; break;
    case ESPADAS: res+="espadas"; break;
    case BASTOS: res+="bastos"; break;
  }
  return res;
}

public boolean winner(Card c, int dominant) {
  if (this.color!=c.color && this.color==dominant) return true;
  if (this.color==c.color) {
    if (this.value==1) return true;
    if (this.value==3 && c.value!=1) return true;
    if (this.value>c.value && c.value!=1 && c.value!=3) return true;
  }
  return false;
}
```

2. Given a `Date` class with the following documentation:



Class Date

java.lang.Object
Date

public class **Date**
extends java.lang.Object

Constructor Summary

Constructors

Constructor and Description
`Date()` Creates date with today's date
`Date(int d, int m, int y)` Creates date with given day, month, and year

Method Summary

Methods

Modifier and Type	Method and Description
int	`getD()` Returns day of the Date object
int	`getM()` Returns month of the Date object
int	`getY()` Returns year of the Date object
java.lang.String	`toString()` Returns date in format "dd/mm/year"
boolean	`validate()` Validates that Date object is correct

Implement a program class that

a) asks for the values of one date,

b) creates the corresponding `Date` object to the inputted day,

c) creates the corresponding `Date` object for today,

d) validate inputted day: in case of invalid date, an error message is produced

e) if inputted day is valid shows if it is previous, the same, or posterior than today, and

f) prints to what season pertains

- Spring is from March 20th to June 20th,
- Summer from June 21st to September 21st
- Autumn from September 22nd to December 21st,
- Winter from December 22nd to March 19th

```
import java.util.*;

public class DateProgram {
  public static void main(String [] args) {
    Scanner kbd=new Scanner(System.in).useLocale(Locale.US);
    int d, m, y;
    Date d1, today;
    String auxDate;

    System.out.print("Date (day, month, year): ");
    d=kbd.nextInt(); m=kbd.nextInt(); y=kbd.nextInt();
    d1=new Date(d,m,y); today=new Date();

    if (d1.validate()) {
      if (d1.getY()>today.getY()) auxDate="posterior";
      else if (d1.getY()<today.getY()) auxDate="previous";
      else if (d1.getM()>today.getM()) auxDate="posterior";
      else if (d1.getM()<today.getM()) auxDate="previous";
      else if (d1.getD()>today.getD()) auxDate="posterior";
      else if (d1.getD()<today.getD()) auxDate="previous";
      else auxDate="the same";  // They are the same date

      System.out.println("Inputted date is "+auxDate+" than today");

      System.out.print("And corresponds to: ");
      if (d1.getM()==1 || d1.getM()==2) System.out.println("Winter");
      if (d1.getM()==3) if (d1.getD()<20) System.out.println("Winter"); else System.out.println("Spring");
      if (d1.getM()==4 || d1.getM()==5) System.out.println("Spring");
      if (d1.getM()==6) if (d1.getD()<21) System.out.println("Spring"); else System.out.println("Summer");
      if (d1.getM()==7 || d1.getM()==8) System.out.println("Summer");
      if (d1.getM()==9) if (d1.getD()<22) System.out.println("Summer"); else System.out.println("Autumn");
      if (d1.getM()==10 || d1.getM()==11) System.out.println("Autumn");
      if (d1.getM()==12) if (d1.getD()<22) System.out.println("Autumn"); else System.out.println("Winter");
    }
    else System.out.println("Incorrect date inputted");
  }
}
```