Name: Sarwan Heer

CodeHS

Coding Task Unit 2

Card.java

```java
public class Card {

    /* Fields */

    /*
        Constants that can be used by other classes to generate
        The card objects
    */

    /* the card has thirteen possible values from 1 to 13 */
    public static final int MAX_VALUE = 13;
    /*
        The card minimum int value of a face forthe card is 80
        which repsents 'P'
    */
    public static final int MIN_FACE = 80;

    /* There are four possible faces */
    public static final int FACES = 4;

    /* value is a number between 1 and 13 inclusive*/
    public int value;

    /*
    *  face is a char with these possible values:
    *   'P', 'Q', 'R', or 'S'
    *  the integer representations of those characters
    *  are 80, 81, 82, and 83 respectively
    */
    public char face;

    /*******************************************************
    * Method name:
    * setFields
    *
    * Description:
    * This method will set the filed attributes value and face
    *
    * Parameters:
    * int - the new value for field value
    * int - the new value for field face
    *
    * Restrictions:
    * value must be between 1 and 13
    * face must be between 80 and 83
    *
    * Return:
    * none
    *
    * Restrictions:
    * none
    *
    * Example: setFields (11, 81) will set:
    * value = 11 and face = 'Q'
    *******************************************************/
    public void setFields(int newValue, int newFace){
        value = newValue;
        face = (char) newFace;
    } // End of setFields method

    /*
       This is the overrided toString method inherited
       from the Object class
    */
    @Override
    public String toString(){
        return "value: " + value + ", face: " + face + "\n";
    } // End of toString method

}
```

```
70. } // End of Card class
```

Library.java

```
 1. import java.util.Random;
 2.
 3. public class Library {
 4.
 5.     /*********************************************************
 6.     * Method name:
 7.     * max
 8.     *
 9.     * Description:
10.     * This method will find the largest number between three
11.     * numbers
12.     *
13.     * Parameters:
14.     * int – first number
15.     * int – second number
16.     * int - third number
17.     *
18.     * Restrictions:
19.     * none
20.     *
21.     * Return:
22.     * int - the largest number
23.     *
24.     * Restrictions:
25.     * Use only Math methods and no conditional statements
26.     *
27.     * Example: max (1,-1, 3) will retun 3
28.     *********************************************************/
29.     public int max (int one, int two, int three){
30.         return Math.max(one, Math.max(two, three));
31.     } // End of max method
32.
33.
34.     /*********************************************************
35.     * Method name:
36.     * generateFromRange
37.     *
38.     * Description:
39.     * This method will generate a random integer between
40.     * two boundaries; low and high, including them
41.     *
42.     * Parameters:
43.     * int – low boundary
44.     * int – high boundary
45.     *
46.     * Restrictions:
47.     * low < high
48.     *
49.     * Return:
50.     * int - a randomly chosen number between low and high
51.     *       inclusive
52.     *
53.     * Restrictions:
54.     * none
55.     *
56.     * Example: generateFromRange(10,15) should return a number
57.     * 10 <= n <= 15
58.     *********************************************************/
59.     public int generateFromRange(int low, int high){
60.         Random rand = new Random();
61.         return rand.nextInt(high - low + 1) + low;
62.     } // End of generateInterval method
63.
64.     /*********************************************************
65.     * Method name:
66.     * pickCard
67.     *
68.     * Description:
69.     * This method will generate and return a Card object
70.     * with randomly chosen fields:
71.     * 1<= value <= 13 and 'P' <= face <= 'S'
72.     *
73.     * Parameters:
74.     * none
75.     *
76.     * Restrictions:
77.     * none
78.     *
79.     * Return:
80.     * Card - A Card object with proper field values
81.     *
82.     * Restrictions:
```

```
    * Do not use arrays or structures
    * not yet covered in this course!!
    * Check the fields, constants, and methods in the
    * Card.java file
    *
    * Example: pickCard() could return a Card object
    * with card.value = 9 and face value = 'P'
    **********************************************************/
    public Card pickCard(){
        Random rand = new Random();
        Card card = new Card();
        card.setFields(rand.nextInt(Card.MAX_VALUE) + 1, rand.nextInt(Card.FACES) + Card.MIN_FACE);
        return card;
    } // End of pickCard method

    /*****************************************************
    * Method name:
    * generateEmail
    *
    * Description:
    * This method will generate an email with the following
    * format:
    * first initial dot lastname @ domain name, all lowercase
    *
    * Parameters:
    * String – full name
    * String – domain name
    *
    * Restrictions:
    * neither parameters are empty
    * the full name can have only one blank space between
    * first and last name
    *
    * Return:
    * String - the email properly formatted
    *
    * Restrictions:
    * none
    *
    * Example: generateEmail("George MurZakU","Domain.Com")
    * should return g.murzaku@domain.com
    **********************************************************/
    public String generateEmail(String fullName, String domain){
        String[] names = fullName.split(" ");
        String firstName = names[0].toLowerCase();
        String lastName = names[1].toLowerCase();
        return firstName.charAt(0) + "." + lastName + "@" + domain.toLowerCase();
    } // End of generateEmail method

    /*****************************************************
    * Method name:
    * getAngle
    *
    * Description:
    * This method will find the angle for a right angle
    * triangle with given opposite and adjacent sides using
    * the inverse of the tan ratio:
    * angle = atan (opposite/adjacent)
    *
    * Parameters:
    * double – opposite side of the given angle
    * double – adjacent side of the angle
    *
    * Restrictions:
    * adjacent cannot be zero
    *
    * Return:
    * double - the rounded angle in degrees to one decimal digit
    *
    * Restrictions:
    * none
    *
    * Example: angle(3, 4) will return 36.9
    **********************************************************/
    public double getAngle(double opposite, double adjacent){
        return Math.toDegrees(Math.atan(opposite / adjacent));
    } //End of getAngle method

} // End of Library class
```

Main.java

```
public class Main
{
    //DO NOT CHANGE OR MODIFY THIS FILE!!
    public static void main(String[] args)
```

```
    {
        Library library = new Library();

        // Testing max method
        System.out.println("calling max(5,3,4)");
        int max = library.max(5,3,5);
        System.out.println(max == 5? "PASS":"FAIL");
        // End of test for max method

        // Testing pickCard method
        System.out.println("calling pickCard()");
        boolean result = true;
        for (int i = 0; i < 100; i++){
            Card card = library.pickCard();
            result = card.value >= 1 && card.value <= 13;
            result = result && (card.face >= 80 && card.face <= 83);
        }
        System.out.println(result ? "PASS" : "FAIL");
        // End of test for pickCard method

        // Testing generateFromRange method
        System.out.println("calling generateFrom(10,20)");
        result = true;
        for (int i = 0; i < 100; i++){
            int number = library.generateFromRange(10,20);
            result = result && (number >= 10 && number <= 20);
        }
        System.out.println(result ? "PASS" : "FAIL");
        // End of test for generateInterval method

        // Testing generateEmail method
        System.out.println("calling generateEmail(\"George Murzaku\", \"Mss.cOm\")");
        String email = library.generateEmail("George MurZaku", "Mss.cOm");
        System.out.println(email.equals("g.murzaku@mss.com")? "PASS" : "FAIL");
        // End of test for generateEmail method

        // Testing getAngle method
        System.out.println("Calling getAngle(3,4)");
        double angle = library.getAngle(3,4);
        System.out.println(Math.abs(angle - 36.9) < 0.2 ? "PASS" : "FAIL");
        // End of test for getAngle method

    } //END OF main METHOD
}//END OF MAIN CLASS
```