

Two Problems

Make sure that you adhere to the below mentioned instructions to implement your classes, do not add any extra information that has not been asked.

Problem 1

Implement an interface named `Colorable` with a void method named `howToColor()`.

Implement an abstract class named `GeometricObject` that has just one abstract method `getArea()` (no other methods, constructors and data fields).

Implement a class named `Square` that extends `GeometricObject` and implements `Colorable`. Implement `howToColor` to display the message "Color all four sides". The `Square` class has a private double data field named `side` with its getter and setter methods. It has a no-arg constructor to create a `Square` with side 0, and another constructor that creates a `Square` with the specified side. It also implements the `getArea()` method to return the area of the square.

Use the following main method.

```
public static void main(String[] args) {
    GeometricObject[] objects = {new Square(2), new Square()};

    for (int i = 0; i < objects.length; i++) {
        System.out.println("Area is " + objects[i].getArea());
        if (objects[i] instanceof Colorable)
            ((Colorable)objects[i]).howToColor();
    }
}
```

Problem 2

Write an abstract superclass encapsulating a Vacation. The class has

- Two private variables - destination and budget
- A constructor that takes two parameters assigns the values to the private data fields
- The two accessor and mutators methods
- An abstract overbudget method returning by how much the vacation is over or under budget
- A `toString` method that returns the values of the two private data fields
- An `equals` method that returns true if the two vacations have the same destination and budget

The vacation class has a non-abstract subclass encapsulating an all-inclusive vacation. This subclass has:

- Three private data fields - brand (such as Four Seasons, Ritz-Carlton, etc), rating (1 to 5) and price

- A constructor that takes five parameters and assigns them to the five data fields
- The three accessor and mutator methods
- An overbudget method that returns by how much the vacation is over or under budget
- A toString method that returns the value of all five data fields for the object
- An equals method that returns true if the destination, budget and price are the same for the two vacations
- Implements the Cloneable interface and the clone method

Use the following Test class.

```
public class Test {
    public static void main( String [] args ) throws CloneNotSupportedException{

        AllInclusiveVacation aiv1 = new AllInclusiveVacation( "Bora Bora", 10000.00, "Four Seasons", 5,
9000.50 );
        System.out.println( "The first all-inclusive vacation is:\n" + aiv1 );

        double money1 = aiv1.overBudget( );
        if ( money1 == 0.0 )
            System.out.println( "It is on budget" );
        else if ( money1 > 0.0 )
            System.out.printf( "It is over budget by %5.2f", money1 );
        else
            System.out.printf( "It is under budget by %5.2f", Math.abs(money1));

        AllInclusiveVacation aiv2 = new AllInclusiveVacation( "Bahamas", 2000.00, "ClubMed", 4, 2049.99 );
        System.out.println( "\nThe second all-inclusive vacation is:\n" + aiv2 );

        double money2 = aiv2.overBudget( );
        if ( money2 == 0.0 )
            System.out.println( "It is on budget" );
        else if ( money2 > 0.0 )
            System.out.printf( "It is over budget by %5.2f", money2 );
        else
            System.out.printf( "It is under budget by %5.2f", Math.abs(money2));

        AllInclusiveVacation aiv3 = (AllInclusiveVacation)aiv1.clone();
        System.out.println( "\nThe third all-inclusive vacation is:\n" + aiv3 );

        if ( aiv1.equals( aiv2 ) )
            System.out.println( "aiv1 and aiv2 are equal" );
        else
            System.out.println( "aiv1 and aiv2 are not equal" );
    }
}
```

```
if ( aiv1.equals( aiv3 ) )  
    System.out.println( "aiv1 and aiv3 are equal" );  
else  
    System.out.println( "aiv1 and aiv3 are not equal" );  
}  
}
```