IIP

Test Unit 5 - Possible solution

Year 2014-2015

Name:

1. Implement a datatype class `Property` that represents the features for properties for a real state agency. You must develope:

   a) Attributes for address (String), type (String, e.g., “Flat”, “House”, “Loft”, . . .), area (in square meters), cost (in euros), and number of rooms.

   b) A constructor that receives address, area, and cost, and initialises type to the empty string and number of rooms to 3.

   c) A constructor that receives the data necessary for initing all the attributes.

   d) The `get` and `set` methods for each attribute.

   e) An `equals` method that overrides the functionality of the method of the `Object` class; you must check the values of all the attributes.

   f) A `toString` method that returns the string in format: “TYPE in ADDRESS of AREA square meters (NROOMS rooms): COST euros”.

   g) A method that returns the cost by square meter of the property.

   h) A method that returns if the property has `n` or more rooms, where `n` is the parameter of the method.

   Note: all the constructors and `set` methods must check that the numeric attributes are positive; in other case, constructors must assign an area of 80.0 square meters, a cost of 100,000.00 euros, and a number os rooms equal to 3; in `set` methods, the corresponding attributes will not be modified.

```
public class Property {
  private String address, type;
  private double area, cost;
  private int nRooms;

  public Property(String a, double ar, double c) {
    address=new String(a);
    if (ar>0) area=ar; else area=80.0;
    if (c>0) cost=c; else cost=100000.0;
    type=new String("");
    nRooms=3;
  }

  public Property(String a, String t, double ar, double c, int n) {
    address=new String(a);
    type=new String(t);
    if (ar>0) area=ar; else area=80.0;
    if (c>0) cost=c; else cost=100000.0;
    if (n>0) nRooms=n; else nRooms=3;
  }

  public String getAddress() { return address; }
  public String getType() { return type; }
  public double getArea() { return area; }
  public double getCost() { return cost; }
  public int getNRooms() { return nRooms; }

  public void setAddress(String a) { address=new String(a); }
  public void setType(String t) { type=new String(t); }
  public void setArea(double a) { if (a>0) area=a; }
  public void setCost(double c) { if (c>0) cost=c; }
  public void setNRooms(int n) { if (n>0) nRooms=n; }

  public boolean equals(Object o) {
    return o instanceof Property &&
           address.equals(((Property) o).address) &&
           type.equals(((Property) o).type) &&
           nRooms==((Property) o).nRooms &&
           area==((Property) o).area &&
           cost==((Property) o).cost;
  }
  public String toString() {
    return type+" in "+address+" of "+area+" square meters ("+nRooms+" rooms): "+cost+" euros";
  }

  public double costBySqMeter() { return cost/area; }

  public boolean hasAtLeastNRooms(int n) { return (nRooms>=n); }

}
```

2. Implement a program class that has a `main` method that asks for the data of two `Property` objects (all data in each property) and calls another `static` method (in the same class) that receives as paramters the two properties and returns the one that has cheaper cost by square meter; this returned property must be shown on the screen in the `main` method.

```
import java.util.*;

public class TestProperty {

  public static void main(String [] args) {
    Scanner kbd=new Scanner(System.in).useLocale(Locale.US);
    Property p1, p2;
    String t, a;
    double c, ar;
    int nr;

    System.out.print("Property 1 - Address: "); a=kbd.nextLine();
    System.out.print("Property 1 - Type: "); t=kbd.nextLine();
    System.out.print("Property 1 - Area: "); ar=kbd.nextDouble();
    System.out.print("Property 1 - Cost: "); c=kbd.nextDouble();
    System.out.print("Property 1 - Number of rooms: "); nr=kbd.nextInt();
    p1=new Property(a,t,ar,c,nr);

    kbd.nextLine();
    System.out.print("Property 2 - Address: "); a=kbd.nextLine();
    System.out.print("Property 2 - Type: "); t=kbd.nextLine();
    System.out.print("Property 2 - Area: "); ar=kbd.nextDouble();
    System.out.print("Property 2 - Cost: "); c=kbd.nextDouble();
    System.out.print("Property 2 - Number of rooms: "); nr=kbd.nextInt();
    p2=new Property(a,t,ar,c,nr);

    System.out.println("Cheapest cost by area in: "+cheaperByArea(p1,p2));
  }

  public static Property cheaperByArea(Property p1, Property p2) {
    if (p1.costBySqMeter()<p2.costBySqMeter()) return p1;
    return p2;
  }
}
```