Programmation Objet Avancée - TP5

—o000o——o000o—

Héritage et abstraction

1 Gestion d'une ville

Dans un souci de transparence envers ses citoyens, une ville décide de concevoir un système informatique lui permettant de représenter l'ensemble des biens (bâtiment, véhicule, . . .) qu'elle possède.

1. Un bien est défini par sa **valeur** et son **coût d'entretien mensuel**. Écrire la classe `Property`

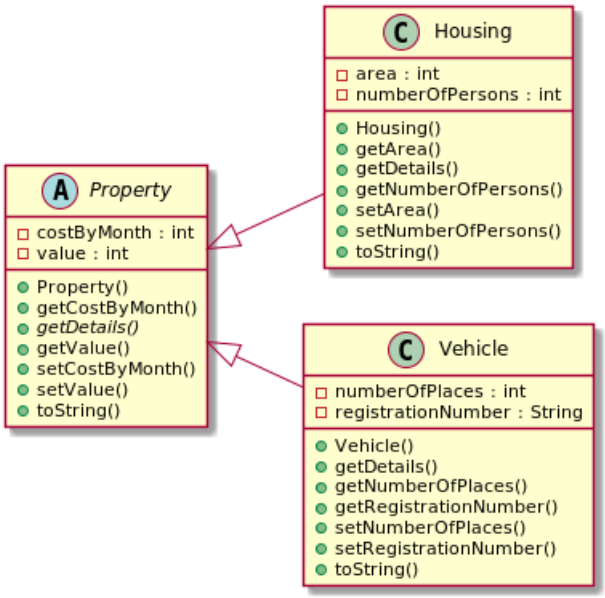
Solution :

```java
package poo.tp5;

public abstract class Property {
    private int value;
    private int costByMonth;

    public Property(int value, int costByMonth) {
        this.value = value;
        this.costByMonth = costByMonth;
    }

    public int getValue() {
        return value;
    }

    public void setValue(int value) {
        this.value = value;
    }

    public abstract String getDetails();

    public int getCostByMonth() {
        return costByMonth;
    }

    public void setCostByMonth(int costByMonth) {
        this.costByMonth = costByMonth;
    }

    @Override
    public String toString() {
        return "Property{" +
            "value=" + value +
            ", costByMonth=" + costByMonth +
            '}';
    }
```

```
}
```

java/Property.java

2. Chaque bien appartient à une catégorie qui lui permet de posséder des informations supplémentaires. Ainsi, les **véhicules** possèdent un **numéro d'immatriculation** et un **nombre de places maximum**, tandis que les **logements** sont caractérisés par leur **superficie au sol** et le **nombre de personnes qu'ils peuvent accueillir**. Écrire les classes permettant de modéliser cette application



Solution :

```java
package poo.tp5;

public class Vehicle extends Property {
    private String registrationNumber;
    private int numberOfPlaces;

    public Vehicle(int value, int costByMonth, String
        registrationNumber, int numberOfPlaces) {
        super(value, costByMonth);
        this.registrationNumber = registrationNumber;
        this.numberOfPlaces = numberOfPlaces;
    }

    public String getRegistrationNumber() {
        return registrationNumber;
    }

    public void setRegistrationNumber(String registrationNumber) {
        this.registrationNumber = registrationNumber;
    }

    public int getNumberOfPlaces() {
```

```
22          return numberOfPlaces;
23      }
24
25      public void setNumberOfPlaces(int numberOfPlaces) {
26          this.numberOfPlaces = numberOfPlaces;
27      }
28
29      @Override
30      public String getDetails() {
31          return "I'm a vehicle";
32      }
33
34      @Override
35      public String toString() {
36          return "Vehicle{" +
37              "registrationNumber='" + registrationNumber + '\'' +
38              ", numberOfPlaces=" + numberOfPlaces +
39              "} " + super.toString();
40      }
41  }
```

java/Vehicle.java

Solution :

```
1   package poo.tp5;
2
3   public class Housing extends Property {
4       private int area;
5       private int numberOfPersons;
6
7       public Housing(int value, int costByMonth, int area, int
            numberOfPersons) {
8           super(value, costByMonth);
9           this.area = area;
10          this.numberOfPersons = numberOfPersons;
11      }
12
13      public int getArea() {
14          return area;
15      }
16
17      public void setArea(int area) {
18          this.area = area;
19      }
20
21      public int getNumberOfPersons() {
22          return numberOfPersons;
23      }
24
25      public void setNumberOfPersons(int numberOfPersons) {
26          this.numberOfPersons = numberOfPersons;
```

```java
    }

    @Override
    public String getDetails() {
        return "I'm a housing";
    }

    @Override
    public String toString() {
        return "Housing{" +
            "area=" + area +
            ", numberOfPersons=" + numberOfPersons +
            "} " + super.toString();
    }
}
```

java/Housing.java

3. Redéfinissez la méthode `String toString()` dans l'ensemble des classes

4. Nous souhaitons rendre impossible l'instanciation d'un bien sans que son type ne soit précisé. Déclarez `Property` en tant que classe abstraite, puis définissez la méthode `String getDetails()`,(chaque détails étant spécifique à un type)

5. Pour le système d'information que nous sommes en train de développer, une ville peut être considérée comme une classe d'objets qui référence l'ensemble des biens qu'elle possède. Cette classe doit offrir les services suivants :

   - Consulter les informations d'un bien particulier
   - Consulter le nombre total de véhicules
   - Calculer le coût total mensuel d'entretien des biens
   - Consulter le nombre total de personnes logées
   - Calculer le coût total mensuel d'entretien pour l'ensemble des véhicules

Solution :

```java
package poo.tp5;

import java.util.List;

public class City {
    private final List<Property> properties;

    public City(List<Property> properties) {
        this.properties = properties;
    }

    public Property findPropertyByIndex(int index) {
        return properties.get(index);
    }

    public long getNumberOfVehicles() {
```

```
17          int numberOfVehicles = 0;
18
19          for (Property property : properties) {
20              if (property instanceof Vehicle) {
21                  numberOfVehicles++;
22              }
23          }
24
25          System.out.println("Old java :: " + numberOfVehicles);
26
27          return properties
28              .stream()
29              .filter(property -> property instanceof Vehicle)
30              .count();
31      }
32
33      public int getMonthlyCost() {
34          int monthlyCost = 0;
35
36          for (Property property : properties) {
37              monthlyCost += property.getCostByMonth();
38          }
39
40          System.out.println("Old java :: " + monthlyCost);
41
42          return properties
43              .stream()
44              .mapToInt(Property::getCostByMonth)
45              .sum();
46      }
47
48      public int getNumberOfPersonsAccommodated() {
49          int numberOfPersons = 0;
50
51          for (Property property : properties) {
52              if (property instanceof Housing) {
53                  numberOfPersons += ((Housing) property).
                        getNumberOfPersons();
54              }
55          }
56
57          System.out.println("Old java :: " + numberOfPersons);
58
59          return properties
60              .stream()
61              .filter(property -> property instanceof Housing)
62              .mapToInt(value -> ((Housing) value).getNumberOfPersons
                    ())
63              .sum();
64      }
65
66      public int getVehicleMonthlyMaintenance() {
67          int costVehicleByMonth = 0;
```

```
68
69          for (Property property : properties) {
70              if (property instanceof Vehicle) {
71                  costVehicleByMonth += property.getCostByMonth();
72              }
73          }
74
75          System.out.println("Old java :: " + costVehicleByMonth);
76
77          return properties
78              .stream()
79              .filter(property -> property instanceof Vehicle)
80              .mapToInt(Property::getCostByMonth)
81              .sum();
82      }
83  }
```

java/City.java

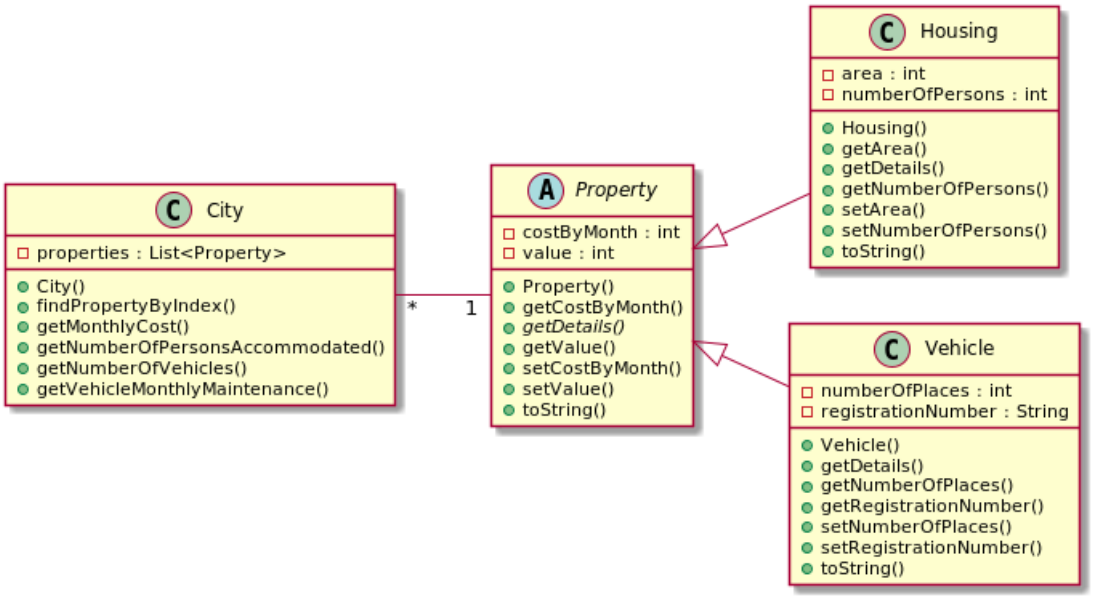
Solution :

```
1  package poo.tp5;
2
3  import java.util.Arrays;
4  import java.util.List;
5
6  public class CityTestCase {
7      public static void main(String[] args) {
8          List<Property> properties = Arrays.asList(
9              new Vehicle(10000, 100, "AB 25 CDF", 5),
10             new Vehicle(12000, 110, "YH 89 UJD", 5),
11             new Housing(500000, 2500, 300, 5),
12             new Housing(125000, 458, 35, 2)
13         );
14
15         City city = new City(properties);
16         System.out.println(city.getNumberOfVehicles());
17         System.out.println(city.getMonthlyCost());
18         System.out.println(city.getNumberOfPersonsAccommodated());
19         System.out.println(city.getVehicleMonthlyMaintenance());
20     }
21 }
```

java/CityTestCase.java

⋆⋆⋆ **Pour aller plus loin**, vous pouvez utiliser les `stream()` offert par Java 8 pour implémenter les méthodes ci-dessus.

Écrivez la classe `City` en vous aidant, notamment, du diagramme de classe ci-dessous.


Housing
area : int
numberOfPersons : int
Housing()
getArea()
getDetails()
getNumberOfPersons()
setArea()
setNumberOfPersons()
toString()
City
properties : List<Property>
City()
findPropertyByIndex()
getMonthlyCost()
getNumberOfPersonsAccommodated()
getNumberOfVehicles()
getVehicleMonthlyMaintenance()
*
1
Property
costByMonth : int
value : int
Property()
getCostByMonth()
getDetails()
getValue()
setCostByMonth()
setValue()
toString()
Vehicle
numberOfPlaces : int
registrationNumber : String
Vehicle()
getDetails()
getNumberOfPlaces()
getRegistrationNumber()
setNumberOfPlaces()
setRegistrationNumber()
toString()