

PROJET JAVA :

Membres :

_MEVENGUE ENGONGOMO FRANCK ANDY

_REMI SINGH

Sommaire :

1.	INTRODUCTION.....	2
1.1.	LES PROBLÈMES RENCONTRÉS	3
1.2.	LES SOLUTIONS TROUVÉES	ERROR! BOOKMARK NOT DEFINED.
2.	LE CODE JAVA.....	6
2.1.	COPIE INTÉGRALE DU CODE COMMENTÉ QUI FONCTIONNE	6
3.	RESULTAT UML	20
4.	RESULTAT JAVA	21
5.	CONCLUSION.....	22

1. INTRODUCTION

Java est un langage de programmation orienté objet et basé sur les classes, il est conçu pour avoir le moins de dépendances d'implémentation possible. Dans ce projet, nous allons essayer de résoudre un problème de la vie courante grâce au langage de programmation java et en utilisant le logiciel starUML pour représenter la situation décrite par le problème à résoudre. Nous avons d'abord utilisé le programme éclipse pour l'implémentation du code et la pratique, nous parlerons des problèmes rencontrés qui seront illustrés à l'aide d'images et les solutions trouvées seront fournies dans le code java à l'aide d'image de l'invite de commande éclipse.

1.1. Les problèmes rencontrés

Erreur 1 : Erreur : la méthode principale est introuvable dans la classe metier.Client, définissez la méthode principale comme suit :

public static void main(String[] args)
ou une classe d'applications JavaFX doit étendre javafx.application.Application

eclipse-workspace - projet_java/src/test/GestionTest.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help



Package Explorer

projet_java

> JRE System Library [JavaSE-1.8]

< src

< dao

> ClientDao.java

< metier

> Client.java

< package2

< Produit

> Test.java

< test

Client.java

ClientDao.java

GestionTest.java

Test.java

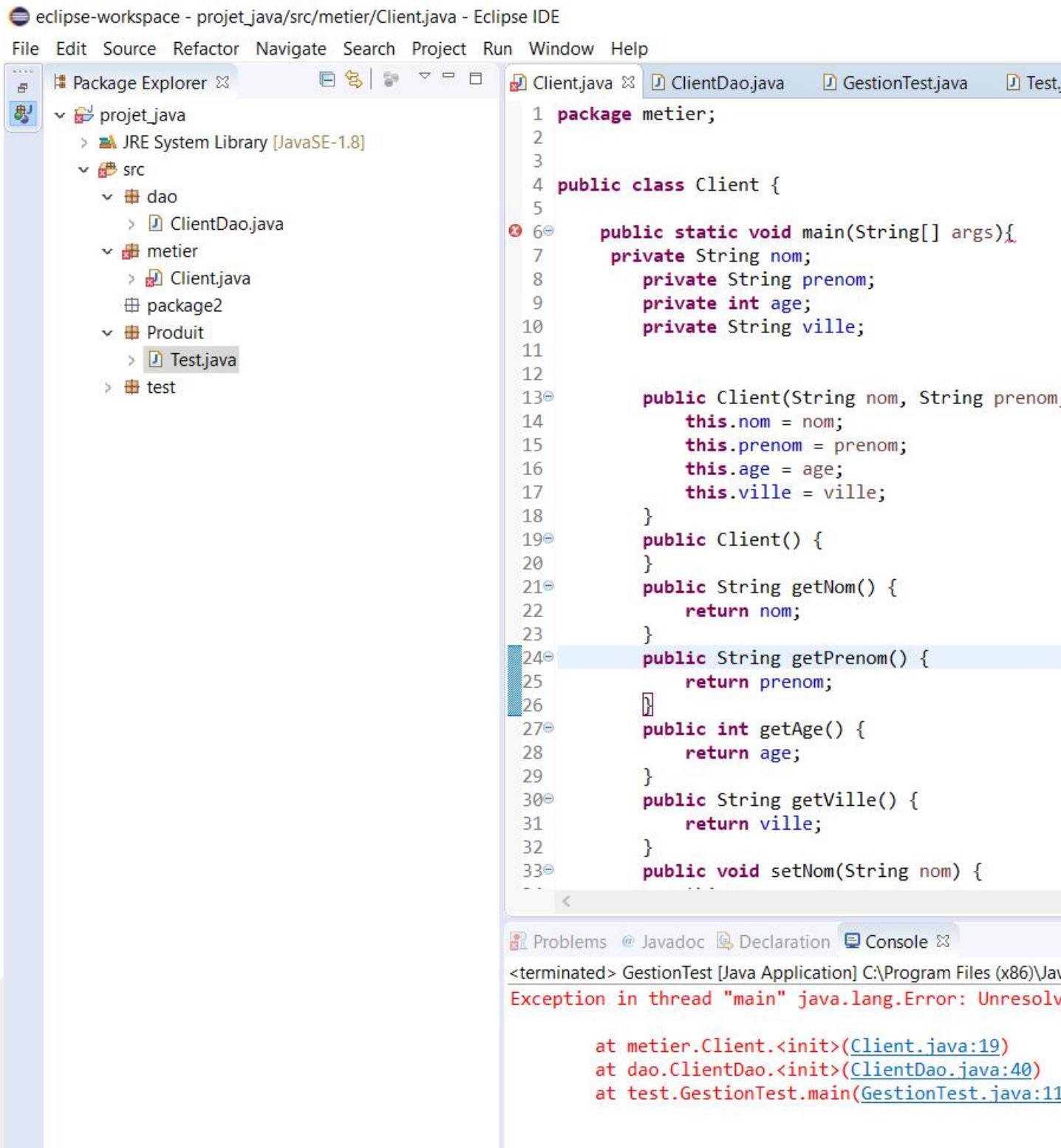
```
1 package test;
2
3 import java.util.ArrayList;
4
5 public class GestionTest {
6     public static void main(String[] argv) {
7         ClientDao clientDao = new ClientDao();
8
9         ArrayList<metier.Client> listeClient = clientDao.selectAll();
10
11        try (Scanner scan = new Scanner(System.in)) {
12            System.out.println("Saisir Mot clé");
13            String mc = scan.next();
14
15            System.out.println(clientDao.delete(mc));
16        }
17
18    }
19
20}
21
22
23
24
25
26
27}
```

Problems @ Javadoc Declaration Console

<terminated> GestionTest [Java Application] C:\Program Files (x86)\Java\

Exception in thread "main" java.lang.Error: Unresolved compilation problems:

```
at metier.Client.<init>(Client.java:19)
at dao.ClientDao.<init>(ClientDao.java:40)
at test.GestionTest.main(GestionTest.java:11)
```

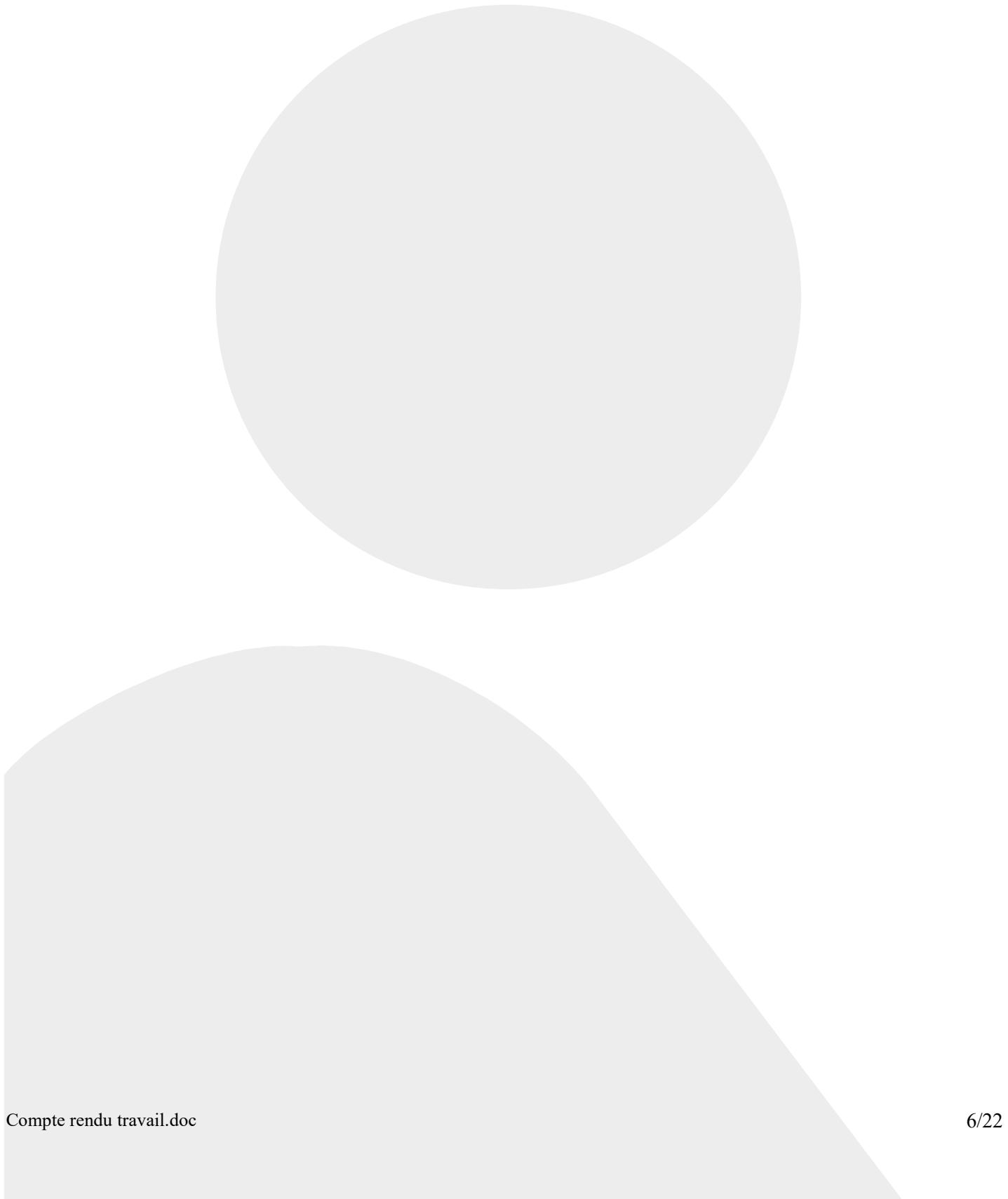


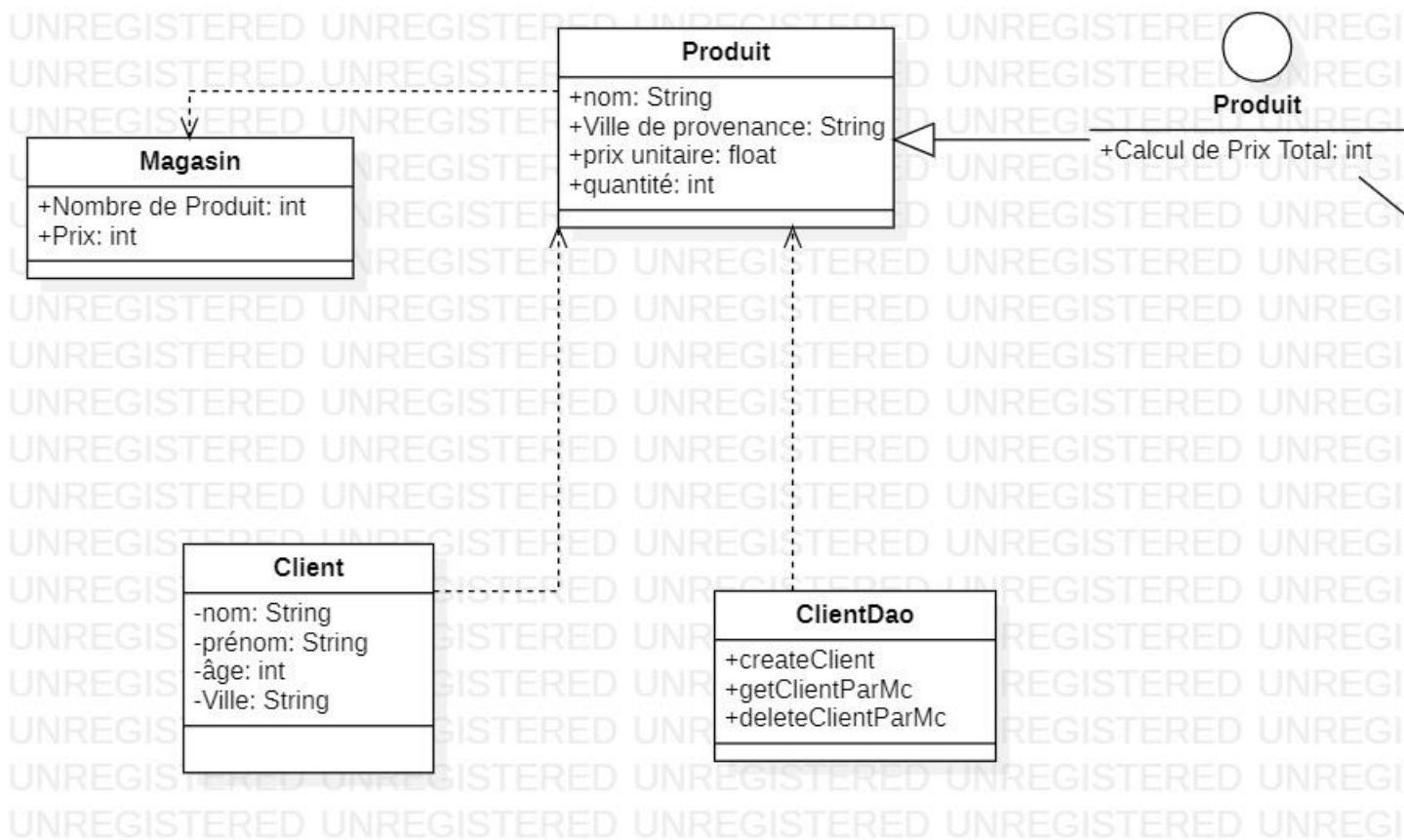
2. LE CODE JAVA

2.1. Copie intégrale du code commenté qui fonctionne

PARTIE -I : UML

1.1 SOLUTION : DIAGRAMME DE CLASSE





PARTIE – II : JAVA

2. Rappel : boucle for each pour les ArrayList d'objet

2.1 Solution : Tableau, Interface graphique, Constructeur

```
1 package Produit;
2
3+import java.awt.GridLayout;□
13
14 public class Test {
15
16+ /*public static JLabel textNom = new JLabel("Nom");
17 public static JLabel textVille = new JLabel("Ville");
18 public static JLabel textQuantite = new JLabel("Quantite");
19 public static JLabel textPrixUni = new JLabel("Prix unitaire (€)");
20
21 public static JTextField nom = new JTextField(10);
22 public static JTextField ville = new JTextField(10);
23 public static JTextField quantite = new JTextField(10);
24 public static JTextField prixUni = new JTextField(10);
25
26 public static JButton valider = new JButton("Valider");*/
27
28+ public static void main(String[] argv){
29
30     JFrame frame = new JFrame();
31     JPanel pan = new JPanel();
32     pan.setLayout(new GridLayout(7, 2));
33
34     JLabel textNom = new JLabel("Nom");
35     JLabel textVille = new JLabel("Ville");
36     JLabel textQuantite = new JLabel("Quantite");
37     JLabel textPrixUni = new JLabel("Prix unitaire (€)");
38     JLabel textQualite = new JLabel("Qualité");
39     JLabel textTauxReduc = new JLabel("Taux de réduction (%)");
40
41     JTextField nom = new JTextField(10);
42     JTextField ville = new JTextField(10);
43     JTextField quantite = new JTextField(10);
44     JTextField prixUni = new JTextField(10);
45     JTextField qualite = new JTextField(10);
46     JTextField tauxReduc = new JTextField(10);
47
48     JButton produitA = new JButton("Produit A");
49     JButton produitB = new JButton("Produit B").
```

```
48     JButton produitA = new JButton("Produit A");
49     JButton produitB = new JButton("Produit B");
50
51     pan.add(textNom);
52     pan.add(nom);
53     pan.add(textVille);
54     pan.add(ville);
55     pan.add(textQuantite);
56     pan.add(quantite);
57     pan.add(textPrixUni);
58     pan.add(prixUni);
59     pan.add(textQualite);
60     pan.add(qualite);
61     pan.add(textTauxReduc);
62     pan.add(tauxReduc);
63     pan.add(produitA);
64     pan.add(produitB);
65
66     frame.setContentPane(pan);
67
68     frame.setSize(400, 200);
69     frame.setLocationRelativeTo(null);
70     frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
71     frame.setVisible(true);
72
73     produitA.addActionListener(new ActionListener(){
74         public void actionPerformed(ActionEvent evnt){
75
76             //JButton boutonClic = (JButton) evnt.getSource();
77             String nom1 = nom.getText();
78             String ville1 = ville.getText();
79             String b = quantite.getText();
80             int quantite1 = Integer.parseInt(b);
81             String a = prixUni.getText();
82             float prixUni1 = Float.parseFloat(a);
83             String qualite1 = qualite.getText();
84
85             ProduitA prod1=new ProduitA(nom1, ville1, prixUni1, quantite1);
86
87             JOptionPane.showConfirmDialog(null, "nom : " + nom1 + ", vil
```

```
92
93    produitB.addActionListener(new ActionListener(){
94        public void actionPerformed(ActionEvent evnt){
95
96            String nom1 = nom.getText();
97            String ville1 = ville.getText();
98            String b = quantite.getText();
99            int quantite1 = Integer.parseInt(b);
100           String a = prixUni.getText();
101           float prixUni1 = Float.parseFloat(a);
102           String c = tauxReduc.getText();
103           int tauxReduc1 = Integer.parseInt(c);
104
105           ProduitB prod2=new ProduitB(nom1, ville1, prixUni1, quantite1);
106
107           JOptionPane.showConfirmDialog(null, "nom : "+nom1+" , ville : "+ville1+" , prix unitaire : "+prixUni1+" , quantité : "+quantite1);
108           JOptionPane.showConfirmDialog(null, "Prix total (avec reduc) : "+(prod2.calculPrixtotal(tauxReduc1)));
109       }
110   });
111 }
112 }
113
114 interface IProduit{
115     public float calculPrixtotal();
116 }
117
118 class ProduitA implements IProduit {
119
120     public String nom;
121     public String ville;
122     public float prixUnitaire;
123     public int quantite;
124     public String qualite;
125
126     public ProduitA(String nom, String ville, float prixUnitaire, int quantite) {
127         this.nom=nom;
128         this.ville=ville;
129         this.prixUnitaire=prixUnitaire;
130         this.quantite=quantite;
131         this.qualite=qualite;
132     }
133 }
```

```
131         this.qualite=qualite;
132     }
133
134     public ProduitA(){}
135
136     public String getNom() {
137         return nom;
138     }
139
140     public String getVille() {
141         return ville;
142     }
143
144     public float getPrixUnitaire() {
145         return prixUnitaire;
146     }
147
148     public int getQuantite() {
149         return quantite;
150     }
151
152     public String getQualite() {
153         return qualite;
154     }
155
156     public void setNom(String nom) {
157         this.nom = nom;
158     }
159
160     public void setVille(String ville) {
161         this.ville = ville;
162     }
163
164     public void setPrixUnitaire(float prixUnitaire) {
165         this.prixUnitaire = prixUnitaire;
166     }
167
168     public void setQuantite(int quantite) {
169         this.quantite = quantite;
```

2.2

```
169         this.quantite = quantite;
170     }
171
172    public void setQualite(String qualite) {
173        this.qualite = qualite;
174    }
175
176    public float calculPrix(){
177        float res = prixUnitaire*quantite;
178        return res;
179    }
180 }
181
182 class ProduitB implements IProduit {
183
184     public String nom;
185     public String ville;
186     public float prixUnitaire;
187     public int quantite;
188     public int tauxReduction;
189
190    public ProduitB(String nom, String ville, float prixUnitaire, int quantite) {
191        this.nom=nom;
192        this.ville=ville;
193        this.prixUnitaire=prixUnitaire;
194        this.quantite=quantite;
195        this.tauxReduction=tauxReduction;
196    }
197
198    public ProduitB() {}
199
200    public String getNom() {
201        return nom;
202    }
203
204    public String getVille() {
205        return ville;
206    }
```

```
~~~  
204    public String getVille() {  
205        return ville;  
206    }  
207  
208    public float getPrixUnitaire() {  
209        return prixUnitaire;  
210    }  
211  
212    public int getQuantite() {  
213        return quantite;  
214    }  
215  
216    public int getTauxReduction() {  
217        return tauxReduction;  
218    }  
219  
220    public void setNom(String nom) {  
221        this.nom = nom;  
222    }  
223  
224    public void setVille(String ville) {  
225        this.ville = ville;  
226    }  
227  
228    public void setPrixUnitaire(float prixUnitaire) {  
229        this.prixUnitaire = prixUnitaire;  
230    }  
231  
232    public void setQuantite(int quantite) {  
233        this.quantite = quantite;  
234    }  
235  
236    public void setTauxReduction(int tauxReduction) {  
237        this.tauxReduction = tauxReduction;  
238    }  
239  
△240    public float calculPrix(){  
241        float res = prixUnitaire*quantite;  
242        return res;  
~~~
```

2.2 Exercice : Client, ArrayList

d'objets, CRUD :

1. Créer le projet final 3 packages : métier , dao , test :

The screenshot shows the Eclipse IDE interface. On the left, the Package Explorer view displays a project named "projet_java" containing a "src" folder with three packages: "dao", "metier", and "test". Inside the "dao" package, there is a file named "ClientDao.java". On the right, the code editor shows the "Client.java" file. The code defines a class "Client" with private attributes "nom", "prenom", "age", and "ville", and corresponding getters and setters. The code editor also shows the status bar at the bottom indicating the application is terminated.

```
private String nom;
private String prenom;
private int age;
private String ville;

public Client(String nom, String prenom, int age, String ville) {
    this.nom = nom;
    this.prenom = prenom;
    this.age = age;
    this.ville = ville;
}

public Client() {}

public String getNom() {
    return nom;
}

public String getPrenom() {
    return prenom;
}

public int getAge() {
    return age;
}

public String getVille() {
    return ville;
}

public void setNom(String nom) {
    this.nom = nom;
}

public void setPrenom(String prenom) {
    this.prenom = prenom;
}

public void setAge(int age) {
    this.age = age;
}

public void setVille(String ville) {
    this.ville = ville;
}
```

2.

eclipse-workspace - projet_java/src/metier/Client.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer Client.java ClientDao.java GestionTest.java

projet_java JRE System Library [JavaSE-1.8]

src

dao ClientDao.java

metier

package2

Produit

test

```
1 package metier;
2
3
4 public class Client {
5
6     public static void main(String[] args)
7     private String nom;
8     private String prenom;
9     private int age;
10    private String ville;
11
12
13    public Client(String nom, String
14        this.nom = nom;
15        this.prenom = prenom;
16        this.age = age;
17        this.ville = ville;
18    }
19    public Client() {
20    }
21    public String getNom() {
22        return nom;
23    }
24    public String getPrenom() {
25        return prenom;
26    }
27    public int getAge() {
28        return age;
29    }
30    public String getVille() {
31        return ville;
32    }
33    public void setNom(String nom) {
34        this.nom = nom;
35    }
36    public void setPrenom(String prenom)
37        this.prenom = prenom;
38    }
39    public void setAge(int age) {
40        this.age = age;
41    }
42}
```

Problems Javadoc Declaration Console

<terminated> Client [Java Application] C:\Program Files (x86)

eclipse-workspace - projet_java/src/metier/Client.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help



Package Explorer

projet_java

> JRE System Library [JavaSE-1.8]
> src
 > dao
 > ClientDao.java
 > metier
 > package2
 > Produit
 > test

Client.java

10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49

ClientDao.java

GestionTest.java

private String ville;

```
    public Client(String nom, String prenom,  
                 this.nom = nom;  
                 this.prenom = prenom;  
                 this.age = age;  
                 this.ville = ville;  
    }  
    public Client() {  
    }  
    public String getNom() {  
        return nom;  
    }  
    public String getPrenom() {  
        return prenom;  
    }  
    public int getAge() {  
        return age;  
    }  
    public String getVille() {  
        return ville;  
    }  
    public void setNom(String nom) {  
        this.nom = nom;  
    }  
    public void setPrenom(String prenom) {  
        this.prenom = prenom;  
    }  
    public void setAge(int age) {  
        this.age = age;  
    }  
    public void setVille(String ville) {  
        this.ville = ville;  
    }  
}
```

Problems @ Javadoc Declaration Console

<terminated> Client [Java Application] C:\Program Files (x86)\Java\jre1

3.

eclipse-workspace - projet_java/src/dao/ClientDao.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

The screenshot shows the Eclipse IDE interface. On the left, the Package Explorer view displays a project structure with a JRE System Library entry and several source files under the 'src' folder, including 'ClientDao.java'. The right side shows the code editor for 'Client.java'. The code implements a DAO class for clients, using a Scanner to read input from standard input and a list to store client objects. The code is annotated with line numbers from 1 to 45. Below the code editor, the status bar shows tabs for Problems, Javadoc, Declaration, Console, and a terminal window indicating the application has terminated.

```

1 package dao;
2
3 import java.util.ArrayList;
4
5
6 public class ClientDao {
7
8     public ArrayList<metier.Client> createClient() {
9
10        try (Scanner scan = new Scanner(System.in)) {
11            System.out.println("Combien de clients ?");
12            int n = scan.nextInt();
13
14            ArrayList<metier.Client> listeClient = new ArrayList<metier.Client>();
15
16            for(int i=0 ; i<n ; i++) {
17                System.out.println("Client n°" + i + " :");
18                System.out.println("saisir nom :");
19                String name = scan.next();
20                System.out.println("saisir prénom :");
21                String firstName = scan.next();
22                System.out.println("saisir prenom :");
23                String lastName = scan.next();
24                System.out.println("saisir age :");
25                int age = scan.nextInt();
26                System.out.println("saisir ville :");
27                String city = scan.next();
28
29                Client client = new Client(name, firstName, lastName, age, city);
30                listeClient.add(client);
31            }
32
33            return listeClient;
34        }
35    }
36
37    Client client = new Client();
38
39    public Client getClientParMC(String mc) {
40        for(int i=0;i<4;i++){
41            if(listeClient.get(i).getNom().equals(mc))
42                client = listeClient.get(i);
43        }
44
45        return client;
46    }
}

```

eclipse-workspace - projet_java/src/dao/ClientDao.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help


Package Explorer

projet_java

> JRE System Library [JavaSE-1.8]

src

> dao

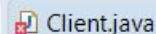
> ClientDao.java

> metier

> package2

> Produit

> test


Client.java

ClientDao.java

GestionTest.java

```

26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

```

```

String firstName = scan.next();
System.out.println("saisir un nom");
int age = scan.nextInt();
System.out.println("saisir un age");
String city = scan.next();
Client client = new Client(firstName, age, city);
listeClient.add(client);
}
return listeClient;
}
Client client = new Client();

public Client getClientParMC(String mc,
for(int i=0;i<4;i++){
if(listeClient.get(i).getNom().equals(mc))
client = listeClient.get(i);
}
return client;
}

public ArrayList<metier.Client> deleteClient(
for(int i=0; i<4; i++){
if(listeClient.get(i).getNom().equals(mc))
listeClient.remove(i);
}
return listeClient;
}
}

```

Problems **@ Javadoc** **Declaration** **Console**

<terminated> Client [Java Application] C:\Program Files (x86)\Java\jre1

4. le package test :

eclipse-workspace - projet_java/src/test/GestionTest.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help



Package Explorer

projet_java

> JRE System Library [JavaSE-1.8]

< src

< dao

ClientDao.java

< metier

package2

< Produit

< test

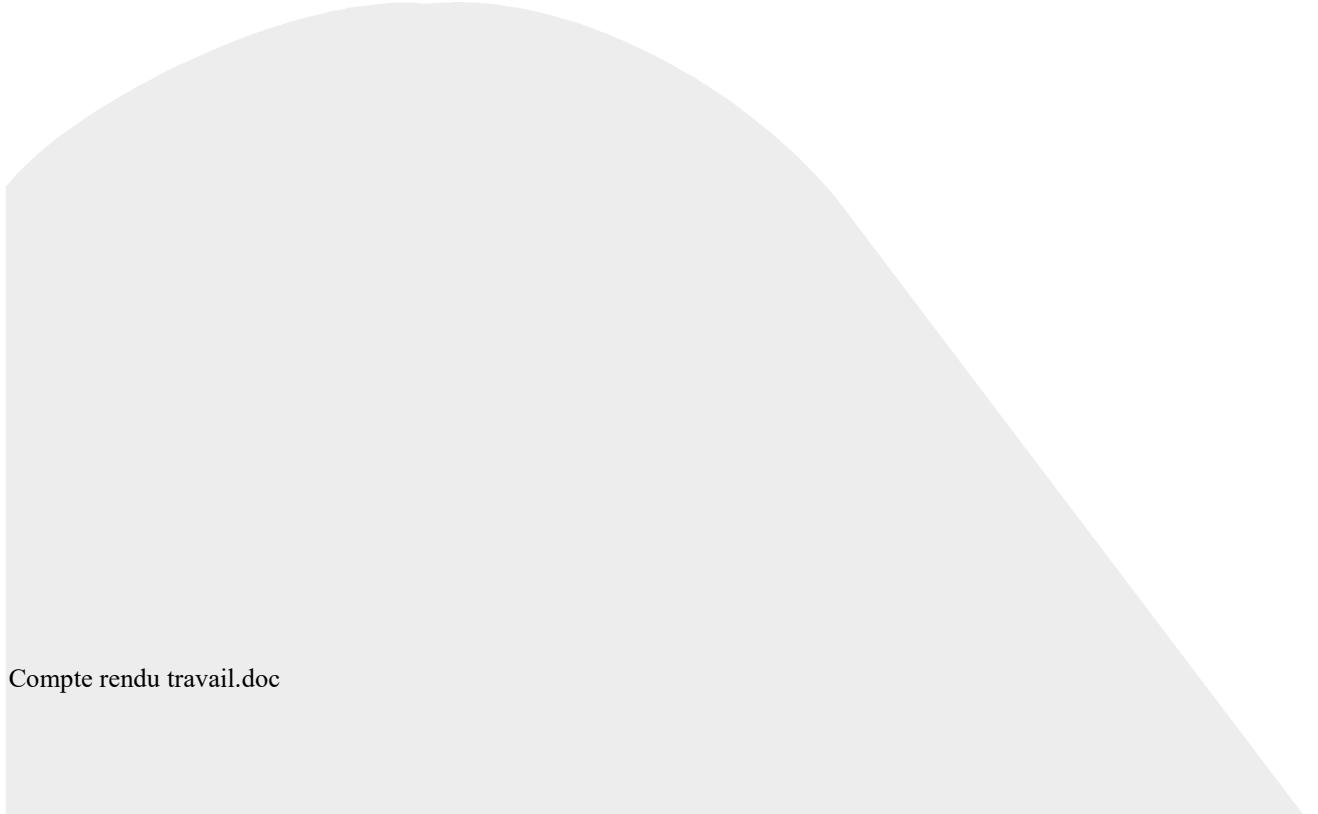
Client.java

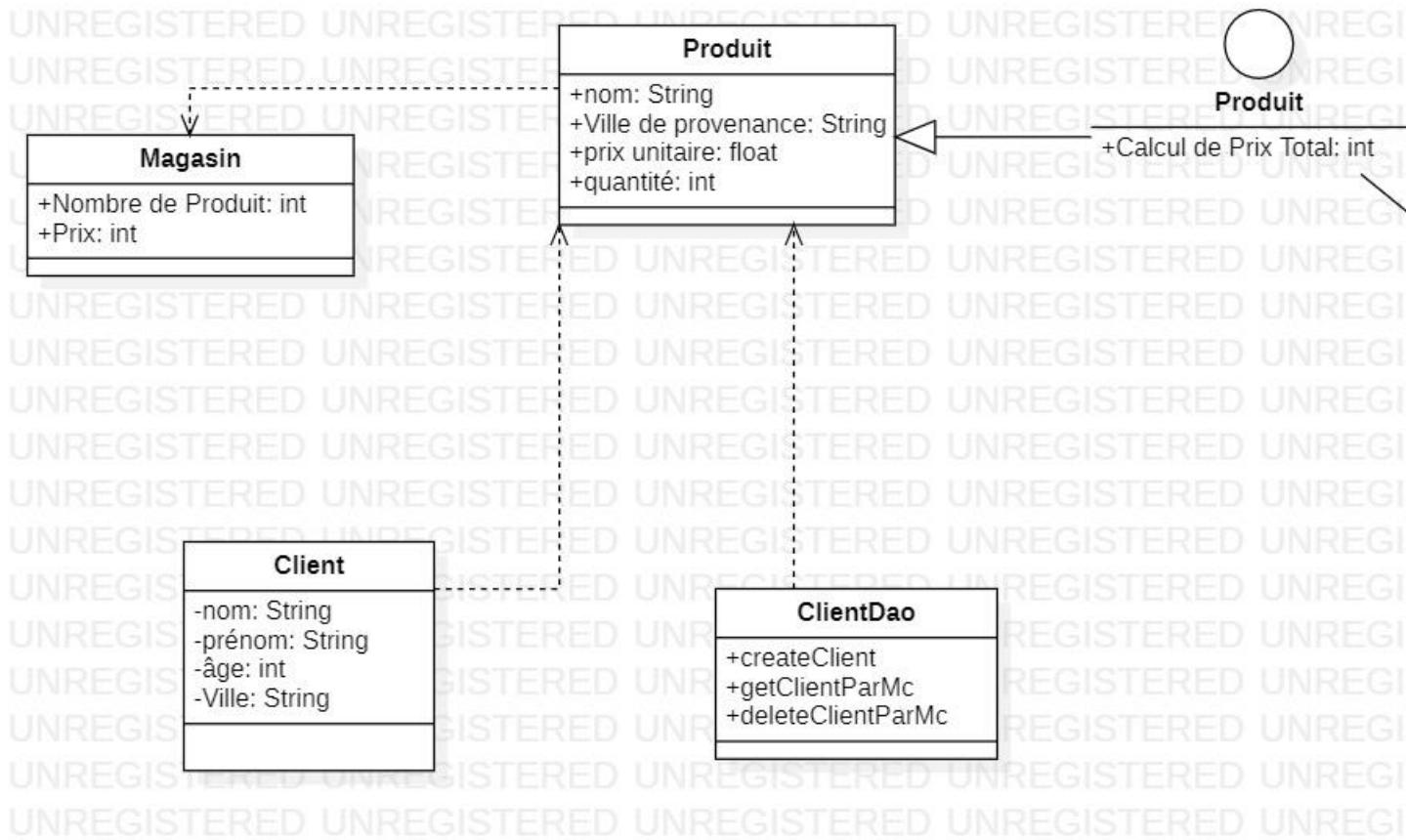
ClientDao.java

GestionTest.java

```
1 package test;
2
3 import java.util.ArrayList;
4
5 public class GestionTest {
6     public static void main(String[] argv) {
7         ClientDao clientDao = new ClientDao();
8
9         ArrayList<metier.Client> listeClient =
10            clientDao.selectAll();
11
12         for (metier.Client client : listeClient) {
13             System.out.println(client);
14         }
15
16         Scanner scan = new Scanner(System.in);
17         System.out.println("Saisir Mot clé");
18         String mc = scan.next();
19
20         System.out.println(clientDao.delete(mc));
21
22     }
23
24 }
```

 Problems @ Javadoc Declaration Console
<terminated> Client [Java Application] C:\Program Files (x86)\Java\jre1

3. RESULTAT UML



4. RESULTAT JAVA

PARTIE – II : JAVA

2. Rappel : boucle for each pour les ArrayList d'objet

2.1 Solution : Tableau, Interface graphique, Constructeur



5. CONCLUSION

Dans ce projet, nous avons utilisé le logiciel starUML pour représenter une situation à l'aide d'un diagramme de classes, tout en implémentant les notions de classe, d'héritage, d'encapsulation, de multiplicité et de stéréotype. Nous avons également créé l'interface Iproduit ainsi que les classes héritages dans la partie I. Dans la partie résultat, nous avons présenté une interface graphique qui permet d'ajouter un produit sous forme de tableau et aussi d'afficher les caractéristiques du produit que l'on veut ajouter comme prix correspondant. Dans la deuxième partie grâce au logiciel éclipse, nous avons créé 3 packages métiers, dao et test. Dans le package dao, nous avons créé la classe clientDao avec deux méthodes CreateClient() et getClientParMc(). Et dans le dernier package de test nous avons créé la classe GestionTest qui fait appel aux méthodes de la classe ClientDao.