Université Mouloud Mammeri de Tizi-Ouzou **2016/2017**
Faculté de Génie Electrique et d'Informatique
Département d'Informatique

Programmation Orientée Objet
2ème Année *Licence* Informatique

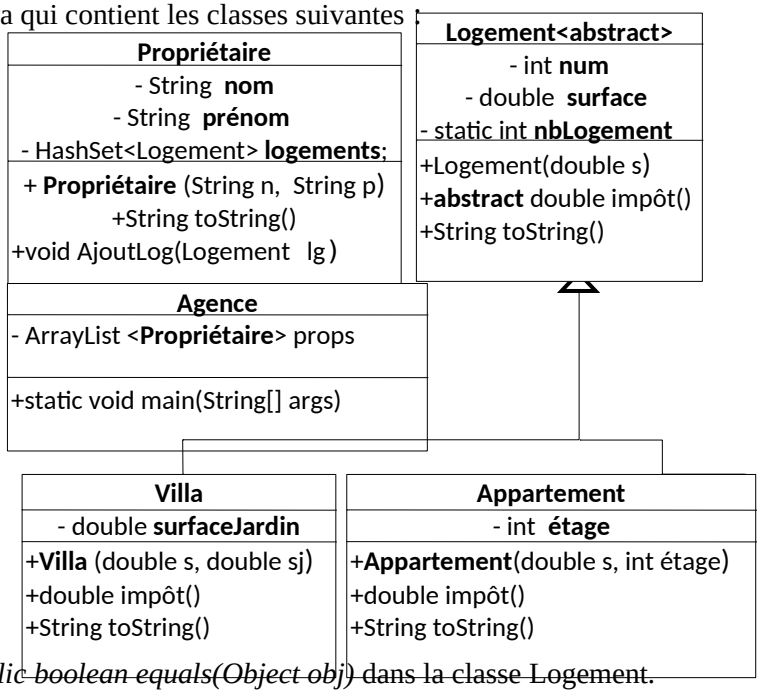
Rattrapage

Questions du cours (5 points):

1. Quelles sont les différences entre une méthode d'instance et une méthode de classe ?
2. Expliquer, brièvement, le fonctionnement du ramasse-miettes (Garbage Collector).
3. Dans quel cas il est recommandé d'utiliser une LinkedList qu'une ArrayList ?, argumenter votre réponse.
4. Expliquer comment un HashSet assure la non duplication de ses éléments.
5. Quelles sont les principales différences entre les bibliothèques AWT et Swing ?

Exercice 1 (8 points):

Ecrire un programme java qui contient les classes suivantes :



Ces classes représentent la gestion des impôts des logements des propriétaires. Un propriétaire peut avoir plusieurs logements (stockés dans un ensemble : HashSet<Logement>). Un logement peut être une villa ou un Appartement.

Le calcul des impôts est réalisé comme suit :
- Pour un **appartement:**
  impôt = surface×(100/(*numéroEtage*+1)).
- Pour une **Villa:**
  impôt = surface×180+ *surfaceJardin*×40.

La méthode AjoutLog(Logement lg) permet d'ajouter un logement pour un propriétaire. Cette méthode lève une exception nommée "**DépassementException**" si le nombre de logements du propriétaire est supérieur à **20**.

Redéfinir les méthodes *public int hashCode()* et *public boolean equals(Object obj)* dans la classe Logement. La méthode *hashCode()* retourne le reste de la division du "***num***" sur 100. La méthode *equals()* retourne vrai si les attributs "***num***" et "***surface***" sont égaux.

Ajouter les fonctions suivantes dans la classe **Agence,** puis appeler les dans la méthode principale ***main***:

- *void AjoutPro(Propriétaire p)* permet d'ajouter un propriétaire dans la structure "***props***".
- *void ajoutLogPro(Propriétaire p, Logement lg)* permet d'ajouter au propriétaire "p" le logement "lg".
- *void trier()* permet de trier la structure "***props***". Le tri est effectué sur la base du nombre de logements des propriétaires. Utiliser pour cela la classe **Collections.**
- *void afficher()* permet d'afficher le contenu de la structure "***props***", en précisant pour chaque propriétaire la valeur de ses impots.

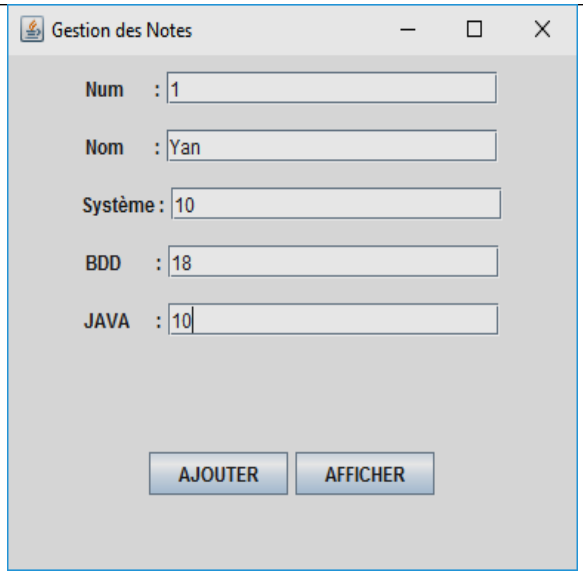
Exercice 2 (7 points) :

Ecrire un programme Java permettant de réaliser l'interface (Figure -1-), comportant :
Cinq zones de texte (**JTextField**), deux boutons (**JButton**): "**Ajouter"** et "**Afficher**" et cinq étiquettes (**JLabel**).

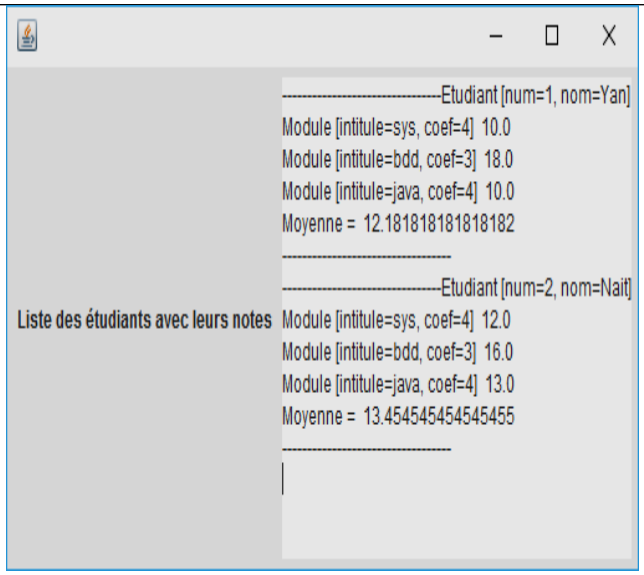
- Le bouton "**Ajouter"** permet d'ajouter un élément dans une structure de type **HashMap<Etudiant, HashMap <Module,Double>>** où la clé est un objet de type **Etudiant,** défini par un ***numéro (int)*** et un ***nom (string),*** et la valeur est une table associative **HashMap <Module,Double>.** Cette table a comme

clé de type **Module,** défini par l'intitulé du module (**String**) "exemples : java, système d'exploitation, etc." et un coefficient (**int**), et comme valeur la note obtenue (**double**)

- Le bouton "**Afficher**" permet d'afficher le contenu de la structure dans une seconde fenêtre illustrée par la figure -2-. Cette fenêtre comporte une étiquette (**JLabel**) et une zone de texte (**JTextArea**).



-Figure 1-



-Figure 2-

Aide-mémoire pour l'exercice 2

```
public class JButton                    // nom de la classe
public  void  addActionListener(ActionListener  al)   //  méthode  d'ajout  d'un
écouteur

public class JLabel
public Label (String titre)

public class JTextField
public JTextArea(int nbCaractères)


public class JTextArea
public JTextArea(int nbLignes, int nbColonnes)
public void append(String  texte)  // ajout d'un texte dans le TextArea

public class JPanel
public JPanel()

public class JFrame
public JFrame(String titre)
public void add(Component comp)  // ajout d'un composant dans un conteneur
public void setLayout(LayoutManager lm)

public interface ActionListener          // interface écouteur
public void actionPerformed(ActionEvent e)  // méthode de l'interface écouteur
```

~ Bon Courage ~