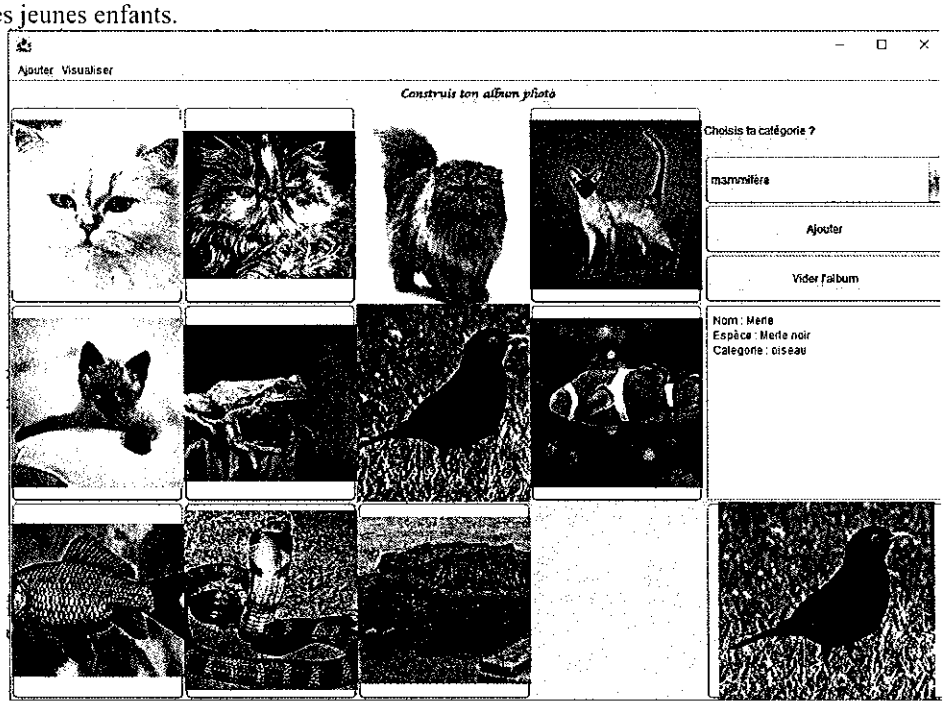
Licence Informatique 1ère année
Module Info2b – Interfaces visuelles

Examen– Durée 2h – Jeudi 12 mai 2022
Documents autorisés (polycopiés de CM, TD et TP)

Un étudiant stagiaire est chargé de développer une application de gestion d'images d'animaux permettant la création d'un album personnalisé, pour des jeunes enfants.



Elle comporte :
- au centre, une galerie de photos disponibles représentant des animaux
- à droite, se trouvent des outils, pour avoir des informations sur les animaux affichés dans la galerie, dont :
  - un bouton « Ajouter » qui permet de mettre la photo de l'animal sélectionné dans l'album. La sélection d'un animal se fait en cliquant sur sa photo dans la galerie. Ses informations textuelles s'affichent dans la zone d'édition et sa photo s'affiche en bas à droite.
  - une liste déroulante (de type JComboBox) qui permet de sélectionner une catégorie d'animal (mammifère, oiseau, reptile, etc) et d'afficher les informations de tous les animaux de cette catégorie dans la zone d'édition (en dessous du bouton « Ajouter »).
  - un bouton « Vider l'album » qui permet de supprimer toutes les photos de l'album qui ont été sélectionnées.

L'application comporte un menu avec une option « Ajouter » avec une sous-option « Animal » qui permet d'ajouter un nouvel animal dans la galerie à l'aide d'une boîte de dialogue, et une option « Visualiser » avec une sous-option « Album » qui permet de visualiser les photos sélectionnées dans l'album à l'aide d'une boîte de dialogue.

Un animal est représenté par sa catégorie (plusieurs animaux peuvent avoir la même catégorie comme « mammifère »), son nom général (par exemple « Chat »), son nom spécifique (par exemple « Persan ») et sa photo qui est de type « ImageIcon ».
Pour gérer les données de cette application, deux classes ont été créées : la classe « Animal » qui modélise un animal et la classe « LesAnimaux » qui permet de gérer l'ensemble des animaux en utilisant une liste de type « ArrayList<Animal> ».

L'annexe 1 détaille la classe « Animal » qui décrit un animal et l'annexe 2 décrit la classe « LesAnimaux » qui permet la gestion de l'ensemble des animaux.

Exercice 1 (3 pts) – *Classes « Animal » et « LesAnimaux »*

En utilisant le code de la classe « Animal » fourni en annexe 1,

a. (1 pt) La classe « Animal » comporte deux accesseurs set (setPhotoF et setPhoto) pour gérer l'affectation de l'attribut photo. Expliquer ces deux accesseurs et leur intérêt.

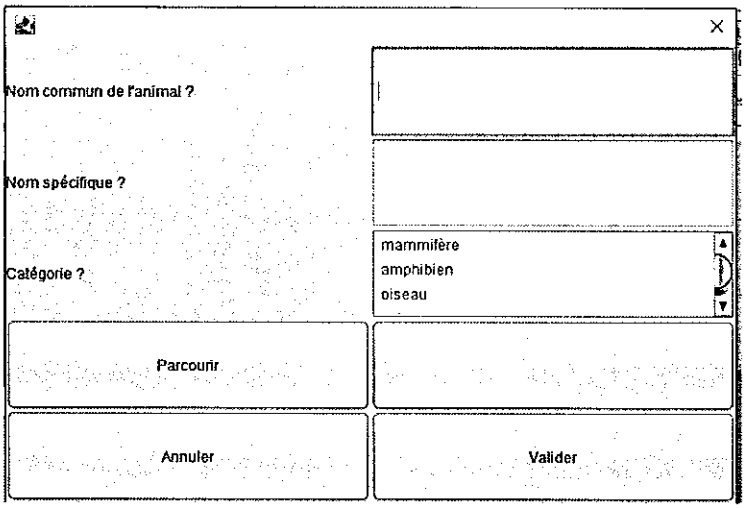
En utilisant le code de la classe « LesAnimaux » fourni en annexe 2,

b. (2 pts) Donner le code java de la méthode « **public void initAnimaux( )** » qui permet l'ajout d'animaux à des fins de tests dans l'application. On pourra par exemple ajouter un animal ayant comme catégorie « mammifère », comme nom général « Chat », comme nom spécifique « Persan », et comme fichier contenant la photo le fichier « chat_persan1.jpg ». On suppose que ce fichier est dans le sous-répertoire « img » du répertoire « src » du projet. Proposer 2 versions de votre code, une en utilisant l'accesseur « setPhotoF », et l'autre en utilisant l'accesseur « setPhoto ».

Exercice 2 (11 pts) – Boite de dialogue d'ajout d'un nouvel animal.

La sélection de la sous-option « Animal » de l'option « Ajouter » du menu, provoque l'ouverture d'une boîte de dialogue permettant la saisie d'un nouvel animal qui pourra être ensuite ajouté dans la galerie.

L'interface de la boite de dialogue nommée « SaisieAnimalDlg » est la suivante :



Elle comporte un composant de type « JList » nommé « ListeCat » pour permettre la sélection de la catégorie de l'animal. Cette liste est remplie à l'ouverture de la boîte de dialogue avec les catégories d'animaux présentés dans la galerie de l'application principale.

Le clic sur le bouton « Parcourir » permet de sélectionner un nom de fichier pour la photo de l'animal.

Le clic sur le bouton « Valider » valide la saisie tandis que le clic sur le bouton « Annuler » permet de l'abandonner.

Si aucune catégorie n'est sélectionnée, par défaut la catégorie « Autre » est affectée à l'animal, et si aucune photo n'est sélectionnée, la photo « ImageDefaut.png » qui se trouve dans le sous-répertoire « img » du répertoire « src » du projet est choisie comme photo par défaut.

a. (1,5 pts) **En utilisant l'image écran de l'application présentée ci-dessus,** proposer une description de l'interface de cette boîte de dialogue sous la forme d'une arborescence avec les types de composant, leur nom et si besoin leur valeur.

b. (1,5 pts) Pour mettre en place cette boîte de dialogue, les attributs suivants ont été retenus :

```
private Animal ani;
private ImageIcon imgAni;
private boolean ok;
```

Expliquer pourquoi selon vous ces attributs ont été choisis pour cette boîte de dialogue et à quoi ils servent.

c. (1 pt) Expliquer quelles sont les informations que doit recevoir la boîte de dialogue pour être fonctionnelle, et comment elles peuvent lui être transmises.

d. (1 pt) Expliquer quelles sont les informations que doit « rendre » la boite de dialogue et sous quelle forme.

e. (1,5 pts) Donner le code complet du constructeur de cette boite de dialogue en commentant vos instructions. Le constructeur devra également remplir la JList « ListeCat » contenant la liste des catégories des animaux de la galerie.

f. (1,5 pts) Donner le code du gestionnaire du clic sur le bouton « Parcourir » en commentant vos instructions. Ce gestionnaire doit permettre la sélection d'un fichier contenant une image, l'affichage de cette image sur le bouton « BPhoto » à droite du bouton parcourir et l'affectation de l'image à l'attribut « imgAni ».

- private void ParcourirActionPerformed(java.awt.event.ActionEvent evt) { ...}

g. (1,5 pts) Donner le code du gestionnaire du clic sur le bouton « Valider » en expliquant en une phrase ce que fait votre gestionnaire et en commentant vos instructions. On vérifiera que deux noms (commun et spécialisé) ont bien été saisis pour valider la saisie.

- private void ValiderActionPerformed(java.awt.event.ActionEvent evt) { ...}

En utilisant l'annexe 3 qui décrit partiellement la classe principale nommée « Examen2022».

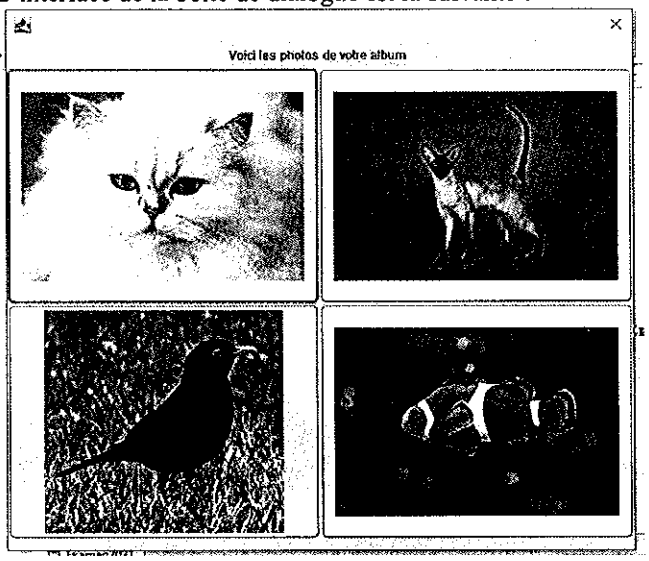
h. (1,5 pts) Donner le code du gestionnaire du clic sur la sous-option « Animal » de l'option « Ajouter » du menu de l'application principale (Examen2022).

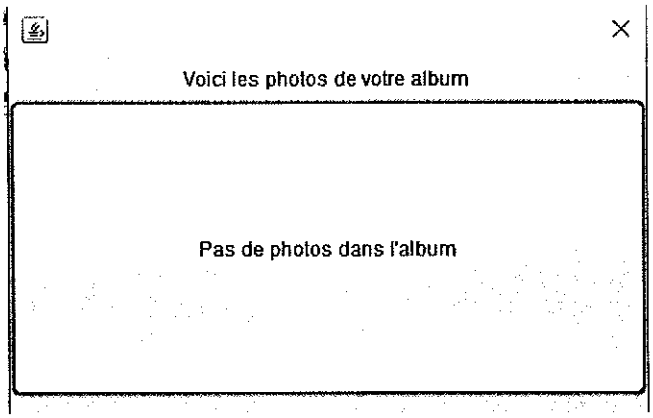
- private void AnimalActionPerformed(java.awt.event.ActionEvent evt) { ...}

Exercice 3 (6 pts) – Boite de dialogue VisuAlbumDlg

La sélection de la sous-option « Album » de l'option « Visualiser » du menu, provoque l'ouverture d'une boîte de dialogue permettant l'affichage sous forme d'une galerie de toutes les photos qui ont été sélectionnées dans la fenêtre principale à l'aide du bouton « Ajouter ».

L'interface de la boîte de dialogue est la suivante :





Elle comporte une galerie des photos présentées sur des boutons s'il existe des photos dans l'album comme ci-dessus si 4 photos sont sélectionnées, ou bien un seul bouton avec un message s'il n'y a pas de photo dans l'album.

Décrire toutes les étapes nécessaires à la mise en place de cette boîte de dialogue en donnant :

a. (1 pt) Une description de cette boite de dialogue sous forme d'un schéma (au choix) indiquant les noms des composants de cette boite de dialogue. On rappelle que la construction des boutons se fait dynamiquement en fonction du nombre de photos de l'album.
b. (1 pt) La description des attributs de la classe en les justifiant.
c. (2 pt) Le code java du constructeur qui devra appeler une méthode « initGalerie() » qui va gérer la construction dynamique des boutons avec les photos en commentant votre code.
d. (2 pt) Le code de la méthode initGalerie().

ANNEXE 1 : Extrait de la classe « Animal »

```
public class Animal {
    private String nomg;  // nom général, par exemple "Chat"
    private String nomp; // nom spécialisé, par exemple "Persan"
    private String categorie; // mammifère, oiseau, reptile, amphibien, poisson
    private ImageIcon photo;


    public Animal(String ng, String np)
    {
      this.nomg=ng;
      this.nomp=np;
      this.categorie="Autre";
      this.setPhoto("ImageDefaut.png");
    }

     public Animal( )
    {
      this.categorie="Autre";
      this.nomg="";
      this.nomp="";
      this.photo = new ImageIcon(getClass().getResource("/img/ImageDefaut.png"));
    }

       public String getNomg() { return this.nomg;}
       public void setNomg(String nomg) {this.nomg = nomg;}
       public String getNomp() {return this.nomp;}
       public void setNomp(String nomp) {this.nomp = nomp;}
       public String getCategorie() {return this.categorie;}
       public void setCategorie(String cat) {this.categorie = cat;}
       public ImageIcon getPhoto() {return this.photo;}

       public void setPhotoF(String fichPhoto) {
                String fich="/img/"+fichPhoto;
                this.photo = new ImageIcon(getClass().getResource(fich));
       }

       public void setPhoto(ImageIcon img) {
                this.photo = img;
        }

      public String toString() {
        String s="";
        s+="Nom : " + this.nomg+ " \n";
        s+="Espèce : " + this.nomp + " \n";
        s+="Categorie : "+ this.categorie+ " \n";
        return s;
       }

}
```

ANNEXE 2 : Extrait de la classe « LesAnimaux »

```
public class LesAnimaux {
   private ArrayList<Animal> lan;

   public LesAnimaux()
   {
      this.lan=new ArrayList<Animal> ();
   }

    public void initAnimaux() { // à compléter    }

    public void ajoutAnimal(Animal t)
   {  this.lan.add(t);   }

   public int getNbAnimaux()
   {  return this.lan.size(); }

   public Animal getAnimal(int ind)
   {
      if (ind >=0 && ind< this.lan.size())
         return this.lan.get(ind);
      else return null;
   }

    public ArrayList<String> getListeCategories()
   {
     ArrayList<String> ll= new ArrayList<String> ();
    for (int i=0; i< lan.size(); i++)
    {   String cat = this.lan.get(i).getCategorie();
        boolean trouve=false;
        for( int j=0; j< ll.size(); j++)
              if (ll.get(j).equals(cat)) trouve=true;
        if (! trouve) ll.add(cat);
     }
     return ll;
   }

   public LesAnimaux getAnimauxCategorie(String cat))
    {   LesAnimaux lp = new LesAnimaux();
               for(int i=0; i<this.lan.size(); i++)
               if(this.lan.get(i).getCategorie().equals(cat))
               lp.ajoutAnimal(this.lan.get(i));
                return lp;}

   public String toString() {
       String s="\n";
      for (int i=0; i<lan.size(); i++)
      {   s+="\n\nAnimal N°"+(i+1)+"\n";
         s+=lan.get(i).toString();
      }
      return s;
    }

}
```

ANNEXE 3 : Extrait de la classe « Examen2022 »

```
public class Examen2022 extends javax.swing.JFrame {

   private LesAnimaux lgal;  // pour gérer tous les animaux de la galerie
   private LesAnimaux lalbum;  // pour gérer les animaux sélectionnés dans l'album
   private Animal aniSelect;  // pour gérer  l'animal sélectionné (dont les informations sont affichées)


   public Examen2022() {
      initComponents();
      this.lgal = new LesAnimaux();
      this.lalbum = new LesAnimaux();
      this.aniSelect=null;
      this.lgal.initAnimaux();
      initGalerie();
      remplirListeCategories();
      Edition.setText("");
   }

private void AnimalActionPerformed(java.awt.event.ActionEvent evt) { //à compléter }


Etc.
```