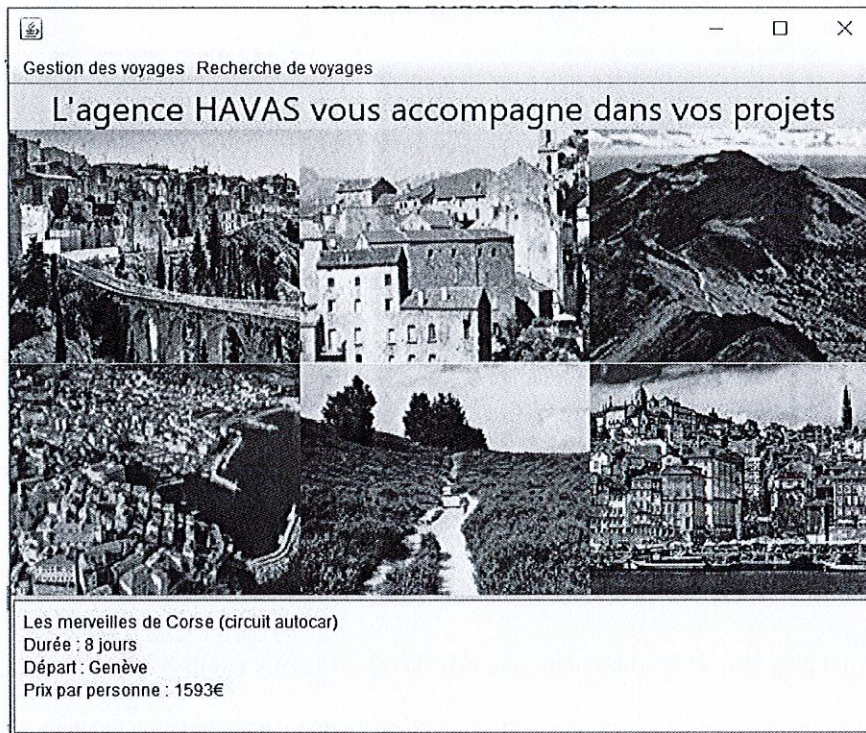


Examen – Durée 2h – Mercredi 22 Mai 2024
Documents autorisés (polycopiés de CM, TD et TP)

Une agence de voyage souhaite proposer une interface à ses clients pour qu'ils sélectionnent leurs voyages. Une personne a été chargée de développer une version enrichie d'une application prototype précédente. L'interface de l'application est la suivante :



Elle comporte :

- Au centre, une galerie de photos des différents voyages proposés, dans un panneau de type « JPanel » nommé « Panneau »
- En bas une zone d'édition (de type « JTextArea » nommée « Edition »), qui permet d'afficher les informations d'un voyage lors du clic sur sa photo.

Deux options de menu sont proposées : l'option « Gestion des voyages » qui comporte une sous-option « Ajouter un nouveau voyage » et l'option « Recherche de voyages » qui a une sous-option « Recherche avancée »

Pour gérer les données de cette application, une classe nommée « Voyage » a été créée. Elle modélise un voyage avec son intitulé, sa ville de départ, sa catégorie qui est un entier (1 pour séjour en hôtel, 2 pour randonnée, 3 pour circuit en autocar), sa durée en nombre de jours, son prix en euros et sa photo. La photo d'un voyage est gérée à l'aide d'une image de type « ImagemIcon ».

L'ensemble des voyages est géré à l'aide d'une classe nommée « LesVoyages » qui comporte une liste de voyages de type « ArrayList<Voyage> ».

Le panneau de la galerie d'images est rempli dynamiquement avec des boutons cliquables. Les boutons sont de type « BoutonImage » qui est une classe qui spécialise la classe « JButton » pour permettre d'avoir un bouton avec une image.

L'annexe 1 décrit la classe « Voyage », l'annexe 2 décrit la classe « LesVoyages » et l'annexe 3 décrit la classe « BoutonImage ».

Exercice 1 (4 pts) :

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

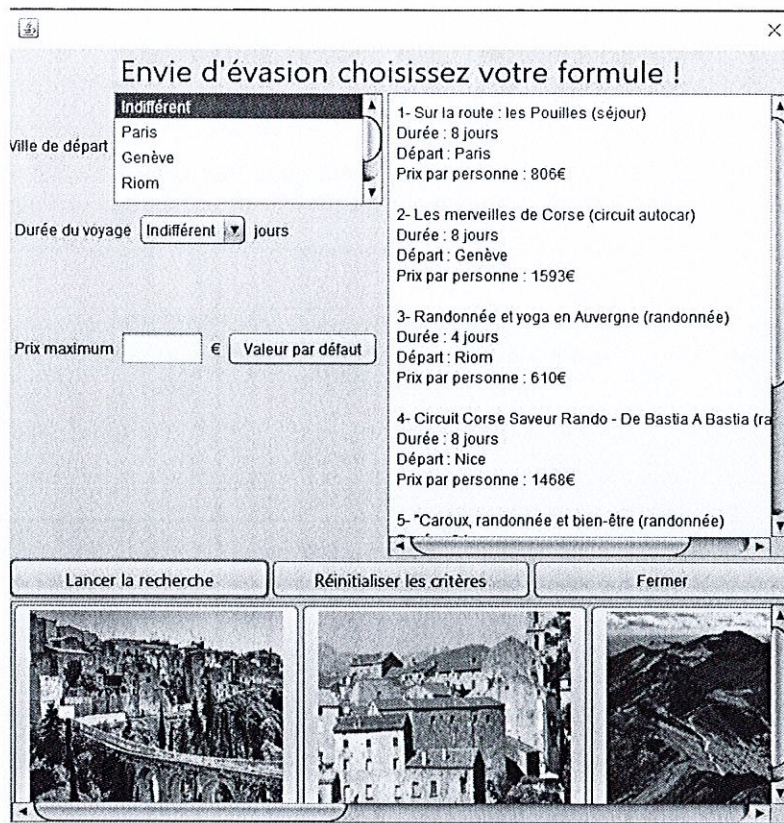
- (1 pt) Expliquer le rôle de la méthode « public LesVoyages getVoyagesPrixMax(int pmax) ».

En utilisant le code de la classe « BoutonImage » fourni en annexe 3,

- (1 pt) Expliquer le rôle de la méthode « paint » et pourquoi elle est nécessaire dans cette classe.
- (1,5 pts) Expliquer en détail le code de cette méthode « paint »
- (0.5 pt) Expliquer quand et comment cette méthode peut être appelée.

Exercice 2 (11 pts) : Boîte de dialogue « RechDlg »

Le clic sur la sous-option « Recherche Avancée » de l’option « Recherche de voyages » ouvre une boîte de dialogue qui permet de rechercher des voyages selon certains critères. L’interface de cette boîte de dialogue est la suivante.



Les critères de recherche sont :

- La ville de départ qui peut être sélectionnée dans une liste de type « JList » nommée « Lvilles » et qui peut avoir la valeur « Indifférent »
- La durée du voyage en nombre de jours, qui peut être sélectionnée dans une liste de type « JComboBox » nommée « CBjours » et qui peut avoir la valeur « Indifférent »
- Le prix maximum qui peut être saisi dans un « JTextField » nommé « PrixMax » et dont la valeur n’est pas considérée si la zone est vide

Le clic sur le bouton « Lancer la recherche » recherche les voyages répondant aux critères sélectionnés et affiche leurs informations textuelles dans la zone d’édition de type « JTextArea » nommée « Edition » et leurs photos dans une galerie construite dynamiquement dans un panneau de type « JPanel » nommé « PanBoutons » en bas de la boîte de dialogue.

Par défaut, si aucun critère n’est sélectionné, tous les voyages sont affichés par le clic sur le bouton « Lancer la recherche ».

En utilisant le code de la classe « RechDlg » partiellement fourni en annexe 4 et son image écran ci-dessus :

- a. (2,5 pts) Proposer **une description partielle de l’interface** de cette fenêtre principale nommée « RechDlg » sous la forme d’une arborescence avec les types de composant, leur nom et si besoin leur valeur, **sans représenter le panneau du bas qui comporte les boutons des actions et la galerie.**
- b. (1,5 pts) Expliquer les paramètres du constructeur de la classe, et pour le 3^{ème} paramètre, pourquoi il est nécessaire, et expliquer précisément les instructions suivantes du constructeur, en les justifiant :

```

this.lstV=lesV;
this.selection=this.lstV;
    
```

- c. (1 pt) La méthode « **private void initDuree()** » permet de remplir la liste, de type « JComboBox » nommée « CBjours », avec la valeur « Indifférent » puis avec les valeurs de 2 à 8 inclus. Donner le code java de cette méthode.
- d. (1,5 pts) La méthode « **private void initVilles()** » permet de remplir la liste, de type « JList », nommée « Lvilles » avec la valeur « Indifférent » puis avec le nom des différentes villes de départ des voyages proposés par l’agence. Donner le code java de cette méthode.

Le gestionnaire d'évènement « `private void LancerActionPerformed(java.awt.event.ActionEvent evt)` » associé au clic sur le bouton « Lancer la recherche » permet d'afficher dans la zone d'édition tous les voyages répondant aux critères sélectionnés et dans la galerie, les photos de ces voyages.

- e. (1,5 pts) Expliquer en une phrase ce que fait la partie de code décrite dans l'annexe au début de la méthode, et détailler ligne à ligne les instructions de cette partie de code.
- f. (3 pts) Selon le même principe, compléter cette méthode
 - Pour filtrer dans la liste « selection » uniquement les voyages de la ville de départ sélectionnée, s'il ne s'agit pas de « Indifférent » (qui est à l'indice 0).
 - Puis pour filtrer selon le prix maximum saisi, si la zone d'édition n'est pas vide.Enfin compléter le code pour qu'il affiche les informations textuelles des voyages de « sélection » dans la zone d'édition, et affiche la galerie de photos correspondante en appelant la méthode « `initPanneauPhotos` ».

Exercice 3 (5 pts) : *Application principale – classe « Examen_2024 » qui dérive de la classe « JFrame »*

En utilisant le code de la classe principale « Examen_2024 » partiellement fourni en annexe 5 ainsi que ceux des classes étudiées précédemment

- a. (1,5 pts) Donner le code du gestionnaire du clic sur la sous-option « Recherche avancée »
« `private void RechVoyagesActionPerformed(java.awt.event.ActionEvent evt)` » qui permet l'ouverture de la boîte de dialogue « RechDlg » en commentant vos instructions, notamment les valeurs des paramètres lors de la création de la boîte de dialogue.

La méthode « `initGalerie()` » de cette classe permet la création et l'affichage de la galerie de photos de tous les voyages du catalogue.

- e. (2 pts) Compléter les 2 lignes manquantes de la méthode « `initGalerie()` » qui réalisent les actions suivantes :
 - Création d'un nouveau bouton image du type « BoutonImage » et nommé « bt »
 - Affection au « Name » du bouton, la valeur de l'indice de la boucle de création des boutonset expliquer précisément les 2 instructions suivantes :
 - `bt.setImage(this.catalogue.getVoyage(i).getPhoto().getImage());`
 - `bt.addActionListener(new java.awt.event.ActionListener() {
 public void actionPerformed(java.awt.event.ActionEvent evt) {
 BtAfficheActionPerformed(evt);
 });`

Le clic sur une photo de la galerie permet l'affichage des informations textuelles du voyage correspondant dans la zone d'édition de type « JTextArea » nommée « Edition ».

- f. (1,5 pts) Donner le code du gestionnaire de clic sur un bouton de la galerie :
« `private void BtAfficheActionPerformed(java.awt.event.ActionEvent evt)` »

ANNEXE 1 : Classe « Voyage »

```
public class Voyage {
    private String intitule;
    private String depart;
    private int categorie; // 1-séjour, 2- randonnée, 3-circuit autocar
    private int duree; // en nombre de jours
    private int prix;
    private ImageIcon photo;

    public Voyage(String titre, int p) {
        this.intitule=titre;
        this.depart="Paris";
        this.duree = 2;
        this.prix=p;
        this.categorie =1;
        this.photo=new ImageIcon(getClass().getResource("/images/logo.png"));
    }

    public Voyage(String titre, String dep, int p) {
        this.intitule=titre;
        this.depart=dep;
        this.duree = 2;
        this.prix=p;
        this.categorie =1;
        this.photo=new ImageIcon(getClass().getResource("/images/logo.png"));
    }

    public String getIntitule() { return this.intitule; }
    public int getCategorie () { return this.categorie; }
    public String getDepart() { return this.depart; }
    public ImageIcon getPhoto() { return this.photo; }
    public int getPrix() { return this.prix; }
    public int getDuree() { return this.duree; }

    public void setIntitule(String s) { this.intitule = s; }
    public void setCategorie(int c) { this.categorie = c; }
    public void setDepart(String dep) { this.depart = dep; }
    public void setPhoto(ImageIcon p) { this.photo = p; }
    public void setPrix(int p) { this.prix = p; }
    public void setDuree(int d) { this.duree = d; }

    @Override
    public String toString()
    {
        String s =this.intitule;
        switch (this.categorie)
        { case 1 : s=s+" (séjour)";break;
          case 2 : s=s+" (randonnée)";break;
          case 3 : s=s+" (circuit autocar)";
        }
        s = s + "\nDurée : "+ this.duree+" jours"
          + "\nDépart : "+this.depart+"\nPrix par personne : "+this.prix+"€";
        return s;
    }
}
```


ANNEXE 2 : Extrait de la classe « LesVoyages »

ATTENTION : Les méthodes de cette classe sont à utiliser dans le code du programme même si le code de celles-ci n'est pas décrit dans l'annexe

```
public class LesVoyages {
    private ArrayList<Voyage> lv;

    public LesVoyages()
    {this.lv = new ArrayList(); }

    public void ajouterVoyage(Voyage s){ this.lv.add(s);}

    public int getNbVoyages() { return this.lv.size(); }

    public Voyage getVoyage(int i){
        if (i>=0 && i<this.lv.size()) return this.lv.get(i); else return null; }

    public LesVoyages getVoyagesPrixMax(int pmax)
    { // à expliquer
        LesVoyages lp = new LesVoyages();
        for(int i=0; i<this.getNbVoyages(); i++)
            if(this.getVoyage(i).getPrix() <= pmax)
                lp.ajouterVoyage(this.getVoyage(i));
        return lp;
    }

    public ArrayList<String> getNomsVillesDepart()
    { // méthode qui retourne la liste des noms des villes de départ sans doublon
    }

    public LesVoyages getVoyagesCategorie(int c)
    { // méthode qui retourne un objet de type « LesVoyages » qui contient tous les voyages
      // dont la catégorie est identique à celle passée en paramètre
    }

    public LesVoyages getVoyagesDuree(int d)
    { // méthode qui retourne un objet de type « LesVoyages » qui contient tous les voyages
      // dont la durée est identique à celle passée en paramètre
    }

    public LesVoyages getVoyagesVilleD(String v)
    { // méthode qui retourne un objet de type « LesVoyages » qui contient tous les voyages
      // dont la ville de départ est identique à celle passée en paramètre
    }

    @Override
    public String toString() {
        String s = "";
        for(int i=0; i<getNbVoyages(); i++)
            s+=(i+1)+"- "+getVoyage(i).toString()+"\n\n";
        return s;
    }

    public LesVoyages(int nb)
    { // constructeur qui permet de créer un certain nombre de voyages déterminé par la valeur passée en paramètre (entre 2 et 6) }
}
}
```

ANNEXE 3 : Classe « BoutonImage »

```
public class BoutonImage extends JButton {
    private Image img;

    public BoutonImage()
    {super(); this.img=null;}

    public BoutonImage(Image i)
    { super(); this.img=i;}

    public Image getImage () { return img;}
    public void setImage (Image i) { this.img=i; repaint(); }

    @Override
    public void paint(Graphics g) {
        super.paint(g);
        if (img != null)
        {
            Image imgB = img.getScaledInstance(this.getWidth(),this.getHeight(), Image.SCALE_DEFAULT);
            this.setIcon(new ImageIcon(imgB));
        }
    }
}
```

ANNEXE 4 : Extrait de la classe « RechDlg »

```
public class RechDlg extends javax.swing.JDialog {
    private LesVoyages lstV; // comporte tous les voyages disponibles
    private LesVoyages selection; // comporte les voyages sélectionnés selon les critères

    public RechDlg(java.awt.Frame parent, boolean modal, LesVoyages lesV) {
        super(parent, modal); initComponents();
        this.lstV=lesV;
        this.selection=this.lstV;
        DefaultListModel mod= new DefaultListModel(); Lvilles.setModel(mod);
        initVilles(); initDuree();
    }
    private void initDuree() { // à coder }
    private void initVilles() { // à coder }

    private void initPanneauPhotos() {
        PanPhotos.removeAll();
        int nb= this.selection.getNbVoyages();
        PanPhotos.setLayout(new GridLayout(1, this.selection.getNbVoyages()));
        for (int i=0; i<nb; i++)
        { JButton bt= new JButton();
          bt.setIcon(this.selection.getVoyage(i).getPhoto());
          PanPhotos.add(bt);
        } }

    private void LancerActionPerformed(java.awt.event.ActionEvent evt) {
        // à expliquer
        this.selection=this.lstV;
        int indD=CBjours.getSelectedIndex();
        if (indD>0)
        { int duree = Integer.parseInt(CBjours.getSelectedItem().toString());
          this.selection = this.selection.getVoyagesDuree(duree);
        }
        // à compléter
    }

    ....
}
```

ANNEXE 5 : Extrait de la classe principale « Examen_2024 » (JFrame)

```
public class Examen_2024 extends javax.swing.JFrame {

    private LesVoyages catalogue; /// pour gérer l'ensemble des voyages

    public Examen_2024() {
        initComponents();
        this.catalogue = new LesVoyages(6); // 6 voyages par défaut
        initGalerie();
    }

    private void initGalerie() {
        int nb= catalogue.getNbVoyages();
        int nblig = (int) Math.sqrt(nb);
        int nbcou = nb/nblig;
        if (nblig*nbcou<nb) nbcou++;
        Panneau.removeAll();
        Panneau.setLayout(new GridLayout(nblig, nbcou));

        for (int i=0; i<nb; i++)
        {
            // instruction 1 à compléter
            // instruction 2 à compléter
            // à expliquer
            bt.setImage(this.catalogue.getVoyage(i).getPhoto().getImage());
            bt.addActionListener(new java.awt.event.ActionListener() {
                public void actionPerformed(java.awt.event.ActionEvent evt) {
                    BtAfficheActionPerformed(evt);}});

            Panneau.add(bt);
        }
    }

    private void BtAfficheActionPerformed(java.awt.event.ActionEvent evt)
    { // à coder }

    private void RechVoyagesActionPerformed(java.awt.event.ActionEvent evt)
    { // à coder }

    ....

}
```