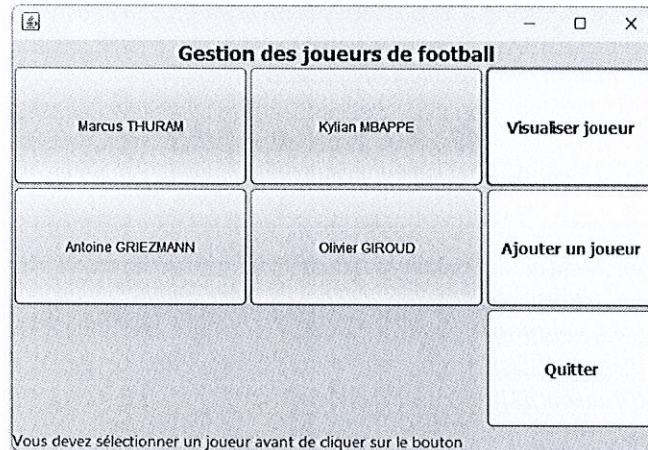


Info2B : Examen session2 – Durée 2h – Jeudi 20 Juin 2024
Documents autorisés (photocopiés de CM, TD et TP)

Un adolescent s'intéresse au football et plus particulièrement aux joueurs français qui jouent en France mais aussi dans les clubs étrangers. Il a développé une application qui permet la gestion d'une liste de joueurs de football dont l'interface est la suivante :



Elle comporte :

- Au centre, une galerie de boutons (de type « JToggleButton ») contenant l'identité des joueurs dans un panneau (de type « JPanel ») nommé « PanJoueurs ». Chacun de ces boutons est sélectionnable et déclenche le même traitement. Un seul de ces boutons peut être sélectionné à la fois.
- A droite, un panneau contenant trois boutons « Visualiser joueur », « Ajouter un joueur » et « Quitter » qui permettent respectivement de visualiser les informations relatives au joueur préalablement sélectionné, d'ajouter un nouveau joueur ou de quitter l'application.
- En bas, une zone de texte (de type « JLabel ») nommée « Message » pour afficher un message en cas d'erreur, notamment quand l'utilisateur clique sur le bouton « Visualiser joueur » alors qu'aucun joueur n'a été préalablement sélectionné.

Pour gérer les données de cette application, une classe nommée « Joueur » a été créée. Elle modélise un joueur avec son nom, son prénom, son poste sur le terrain, le pays où il joue (de type « Pays ») et sa photo. La photo d'un joueur est gérée à l'aide d'une image de type « ImageIcon ». Elle se trouve dans le dossier « src » du projet.

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

Une classe « Pays » a été créée. Elle représente chaque pays par son nom, les trois couleurs de son drapeau ainsi que le sens des couleurs. Ce dernier est 'H' pour des couleurs placées horizontalement sur le drapeau et 'V' pour des couleurs représentées verticalement.

L'annexe 1 décrit partiellement la classe « Joueur », l'annexe 2 décrit la classe « Pays » et l'annexe 3 décrit partiellement la classe « LesJoueurs ».

Exercice 1 (4 pts) : Classes « Joueur » et « LesJoueurs »

En utilisant le code de la classe « Joueur » fourni en annexe 1, et le code de la classe « LesJoueurs » fourni en annexe 3

- (1 pt) Expliquer, en 1 ou 2 phrases, ce que fait le constructeur avec paramètres de la classe « Joueur ».
- (1,5 pts) Rédiger le code de la méthode « public LesJoueurs getJoueurs(Pays p) » de la classe « LesJoueurs » qui prend en paramètre une instance de la classe « Pays » et renvoie la liste des joueurs qui jouent dans ce pays.
- (1,5 pts) Compléter le code de la méthode « public void initJoueurs() » afin d'ajouter, à des fins de tests, le joueur attaquant Kylian MBAPPE qui joue en France. La photo de ce joueur se trouve dans le fichier « km.png » situé dans le dossier « src » du projet.

Exercice 2 (5 pts) : Boîte de dialogue « VisuJoueurDlg »

Le clic sur le bouton « Visualiser joueur » de l'application principale ouvre une boîte de dialogue « VisuJoueurDlg » qui permet de visualiser les informations du joueur choisi.

En utilisant le code de la méthode « initComponents() » fourni en annexe 4,

- a) (2,5 pts) Proposer une description de l'interface de cette fenêtre nommée « VisuJoueurDlg » sous la forme d'une arborescence avec les types de composant, leur nom et si besoin leur valeur. On rappelle que par défaut la stratégie est FlowLayout.

La classe « VisuJoueurDlg » qui dérive de la classe « JDialog » ne possède aucun attribut. Son constructeur est le suivant :

```
public VisuJoueurDlg(java.awt.Frame parent, boolean modal, Joueur j) {
    super(parent, modal);
    initComponents();
    Ltitre.setText(j.getPrenom()+" "+j.getNom()+" (" +j.getPoste()+")");
    Bphoto.setIcon(j.getPhoto());
    Pays pc = j.getPays();
    Info.setText("Drapeau de "+pc.getNom());
    Drapeau.setLayout(new GridLayout(1, 3)); // Disposition verticale des couleurs du drapeau
    Couleur1.setBackground(pc.getDrapeau()[0]);
    Couleur2.setBackground(pc.getDrapeau()[1]);
    Couleur3.setBackground(pc.getDrapeau()[2]);
}
```

- b) (1 pt) Justifier la présence du 3ème paramètre dans le constructeur de la classe « VisuJoueurDlg ».

La fenêtre « VisuJoueurDlg » comporte un panneau, de type « JPanel » nommé « Drapeau », constitué de trois boutons. Ces derniers sont destinés à être coloriés avec chacune des trois couleurs du drapeau correspondant au pays dans lequel le joueur joue. La stratégie du panneau « Drapeau » est modifiée dynamiquement dans le constructeur.

- c) (1,5 pts) Expliquer les 4 dernières instructions du constructeur de la classe « VisuJoueurDlg » et les modifier afin que les trois couleurs du drapeau du joueur passé en paramètre s'affichent dans le bon sens (horizontalement ou verticalement).

Exercice 3 (8,5 pts) : Boîte de dialogue « SaisieJoueurDlg »

Le clic sur le bouton « Ajouter un joueur » de l'application principale ouvre une boîte de dialogue « SaisieJoueurDlg » qui permet de saisir les informations d'un nouveau joueur. L'interface de cette boîte de dialogue est la suivante :

- Deux zones de texte (de type « JTextField »), nommées « Nom » et « Prenom » pour saisir respectivement le nom et le prénom du joueur de football.
- Une liste déroulante (de type « JComboBox »), nommée « CBposte » pour sélectionner le poste du joueur parmi les trois valeurs fournies dans l'EDI : Attaquant, Défenseur et Milieu.
- Une liste (de type « JList »), nommée « Lpays » renseignée par le constructeur à partir de la liste des pays possibles initialisée dans l'application principale.
- Un bouton (de type « JButton »), nommé « Parcourir » qui permet de sélectionner le nom d'un fichier parmi ceux présents dans le dossier « src » du projet.
- Une zone de texte (de type « JLabel »), nommée « Photo », contenant le nom du fichier correspondant à la photo sélectionnée. Par défaut la valeur « default.png » est placée dans cette étiquette (JLabel).
- Deux boutons (de type « JButton »), nommés « Annuler » et « Ajouter », pour respectivement fermer la fenêtre sans tenir compte des valeurs saisies ou fermer la fenêtre en ayant créé une nouvelle instance de « Joueur » avec les différentes valeurs saisies.

En utilisant le code de la classe « SaisieJoueurDlg » partiellement fourni en annexe 5,

- (2 pts) Commenter chaque ligne du code du constructeur de la classe « SaisieJoueurDlg » en expliquant le rôle des méthodes appelées.
- (1,5 pts) Rédiger le code java de la méthode « private void initPays() » qui remplit la liste, de type « JList » nommée « Lpays », avec les noms des pays contenus dans le tableau « lesPays ».
- (1 pt) Expliquer en une phrase le contenu de ce qui est retourné par la méthode « private Pays rechPays(String nomP) » de la classe « SaisieJoueurDlg ».
- (2 pts) Expliquez, en quelques phrases, le code java du gestionnaire de clic sur le bouton « Parcourir » nommé « ParcourirActionPerformed(java.awt.event.ActionEvent evt) » présent dans la classe « SaisieJoueurDlg ».
- (2 pts) Le gestionnaire d'événement du clic sur le bouton « Ajouter » s'assure que des valeurs ont été saisies pour le nom et le prénom du joueur. Dans ce cas, il crée une nouvelle instance de « Joueur », récupère les valeurs saisies par l'utilisateur pour les affecter à l'attribut « jc » de type « Joueur », met le booléen « ok » à vrai et rend la fenêtre non visible. Dans le cas contraire, il ne fait rien. Compléter le code java du gestionnaire « private void BajouterActionPerformed(java.awt.event.ActionEvent evt) » pour qu'il réalise les tâches détaillées ci-dessus.

Exercice 4 (2,5 pts) : : Application principale – classe « Session2_2024 » qui dérive de la classe « JFrame »

L'application principale est gérée par une classe appelée « Session2_2024 » qui dérive de la classe « JFrame ». Cette classe comporte 3 attributs privés :

```
private LesJoueurs lesJoueurs; // liste des joueurs de football
private Pays lesPays[]; // liste des pays dans lesquels les joueurs de football jouent
private Joueur jc; // joueur sélectionné pour lequel on veut visualiser les informations
```

- (1 pt) Expliquer, en une ou deux phrases, le code java du gestionnaire de clic sur le bouton « Visualiser joueur » donné ci-dessous.

```
private void BvisualiserActionPerformed(java.awt.event.ActionEvent evt) {
    if (jc != null)
    { VisuJoueurDlg diag = new VisuJoueurDlg(this, true, jc);
      diag.setVisible(true);
    }
    else
      Message.setText("Vous devez sélectionner un joueur avant de cliquer sur le bouton");
}
```

- (1,5 pts) Donnez le code du gestionnaire associé au bouton « Ajouter un joueur » qui :
 - efface le contenu du composant « Message » de type « JLabel »,
 - permet l'ouverture de la boîte de dialogue de saisie « SaisieJoueurDlg »,
 - et si celle-ci a été fermée en cliquant sur le bouton « Ajouter », ajoute le joueur saisi à la liste des joueurs et appelle la méthode initPanneau() pour mettre à jour le panneau central avec l'identité des joueurs.L'entête du gestionnaire est le suivant :

private void BajouterActionPerformed(ActionEvent evt)

ANNEXE 1 : Classe « Joueur »

```
public class Joueur {
    private String nom, prenom, poste;
    private Pays pays;
    private ImagemIcon photo;

    public Joueur()
    {   this.nom=""; this.prenom=""; this.poste="";
        this.pays=new Pays("France", Color.blue, Color.white, Color.red, 'V');
        this.photo=new javax.swing.ImageIcon(getClass().getResource("default.png")); }

    public Joueur(String n, String pr, String po, Pays pa)
    {   this.nom=n; this.prenom=pr; this.poste=po; this.pays=pa;
        this.photo=new javax.swing.ImageIcon(getClass().getResource("default.png")); }

    public String getNom() { return this.nom; }
    public String getPrenom() { return this.prenom; }
    public String getPoste() { return this.poste; }
    public Pays getPays() { return this.pays; }
    public ImagemIcon getPhoto() { return this.photo; }

    public void setNom(String n) { this.nom = n; }
    public void setPrenom(String p) { this.prenom = p; }
    public void setPoste(String po) { this.poste = po; }
    public void setPays(Pays pa) { this.pays = pa; }
    public void setPhoto(ImagemIcon img) { this.photo = img; }

    public String toString()
    { return this.nom + " " + this.prenom+ " est " + this.poste + " (" +this.pays.getNom()+)"; }
}
```

ANNEXE 2 : Classe « Pays »

```
public class Pays {
    private String nom;
    private Color drapeau[]; // les 3 couleurs du drapeau
    private char sens; // H pour couleurs horizontales, V pour couleurs verticales

    public Pays(String n, Color coul1, Color coul2, Color coul3, char s )
    {   this.nom = n;   this.drapeau = new Color[3];
        this.drapeau[0]=coul1;   this.drapeau[1]=coul2;   this.drapeau[2]=coul3;
        this.sens= s;
    }

    public String getNom() { return nom; }
    public Color[] getDrapeau() { return drapeau; }
    public char getSens() { return sens; }

    public void setNom(String nom) { this.nom = nom; }
    public void setDrapeau(Color[] drapeau) { this.drapeau = drapeau; }
    public void setSens(char sens) { this.sens = sens; }

    public boolean equals(Pays autre) { return this.nom.equals(autre.nom);}
}
```

ANNEXE 3 : Extrait de la classe « LesJoueurs »

```
public class LesJoueurs {
    private ArrayList<Joueur> lstJ;

    public LesJoueurs()
    {   this.lstJ=new ArrayList<>(); }

    public Joueur getJoueur(int i)
    { if (0<=i && i<this.lstJ.size())
        return this.lstJ.get(i);
      else return null;
    }

    public int getNbJoueurs(){ return this.lstJ.size();}
    public void ajouteJoueur(Joueur j) { this.lstJ.add(j);}

    // renvoie le joueur de nom « n » et prénom « p » s'il existe et null sinon
    public Joueur rechJoueur(String n, String p)
    {   Joueur j=null;
        for(int i=0; i<this.getNbJoueurs(); i++)
            if (this.lstJ.get(i).getNom().equals(n) && this.lstJ.get(i).getPrenom().equals(p)))
                j= this.lstJ.get(i);
        return j;
    }

    // initialise la liste des joueurs
    public void initJoueurs()
    {   Pays france = new Pays("France", Color.blue, Color.white, Color.red, 'V');
        // à compléter
    }

    // renvoie la liste des joueurs français jouant dans le pays passé en paramètre
    public LesJoueurs getJoueurs(Pays p)
    {
        // à coder
    }

    public String toString()
    {   String res="";
        for(int i=0; i<this.getNbJoueurs();i++)
            res+=this.getJoueur(i)+"\n";
        return res;
    }
}
```

ANNEXE 4 : Méthode « initComponents() » de la classe « VisuJoueurDlg »

```
private void initComponents() {  
  
    JPanel1 = new javax.swing.JPanel();  
    Bphoto = new javax.swing.JButton();  
    JPanel2 = new javax.swing.JPanel();  
    Info = new javax.swing.JLabel();  
    Drapeau = new javax.swing.JPanel();  
    Couleur1 = new javax.swing.JButton();  
    Couleur2 = new javax.swing.JButton();  
    Couleur3 = new javax.swing.JButton();  
    Ltitre = new javax.swing.JLabel();  
    Bfermer = new javax.swing.JButton();  
  
    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);  
  
    JPanel1.setLayout(new java.awt.GridLayout(1, 2));  
    JPanel1.add(Bphoto);  
  
    JPanel2.setLayout(new java.awt.BorderLayout());  
  
    Info.setFont(new java.awt.Font("Segoe UI", 2, 14));  
    Info.setForeground(new java.awt.Color(0, 0, 255));  
    Info.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);  
    Info.setText("Drapeau de ");  
    JPanel2.add(Info, java.awt.BorderLayout.NORTH);  
  
    Drapeau.setLayout(new java.awt.GridLayout(1, 0));  
    Drapeau.add(Couleur1);  
    Drapeau.add(Couleur2);  
    Drapeau.add(Couleur3);  
  
    JPanel2.add(Drapeau, java.awt.BorderLayout.CENTER);  
  
    JPanel1.add(JPanel2);  
  
    getContentPane().add(JPanel1, java.awt.BorderLayout.CENTER);  
  
    Ltitre.setFont(new java.awt.Font("Tahoma", 1, 18));  
    Ltitre.setText("");  
    getContentPane().add(Ltitre, java.awt.BorderLayout.NORTH);  
  
    Bfermer.setText("Fermer");  
    Bfermer.addActionListener(new java.awt.event.ActionListener() {  
        public void actionPerformed(java.awt.event.ActionEvent evt) {  
            BfermerActionPerformed(evt);  
        }  
    });  
    getContentPane().add(Bfermer, java.awt.BorderLayout.SOUTH);  
  
}
```

ANNEXE 5 : Extrait de la classe « SaisieJoueurDlg »

```
public class SaisieJoueurDlg extends javax.swing.JDialog {

    private Joueur jc;           // joueur courant
    private boolean ok;         // booléen pour savoir si la saisie est complète ou non
    private Pays[] lesPays;     // tableau des pays possibles pour un joueur
    private Imagelcon img;      // image pour la photo du joueur

    public SaisieJoueurDlg(java.awt.Frame parent, boolean modal, Pays[] lesP) {
        super(parent, modal);
        initComponents();
        this.jc=null;
        this.ok=false;
        this.lesPays=lesP;
        initPays();
        this.img = new Imagelcon(getClass().getResource("/default.png"));
        Photo.setText("default.png");
        CBposte.setSelectedIndex(0);
        Lpays.setSelectedIndex(0);
    }

    public boolean getOk() { return this.ok; }
    public Joueur getJoueur(){ return this.jc; }

    private void initPays() // initialise la JList avec le contenu de l'attribut lesPays
    { // à coder }

    private Pays rechPays(String nomP) // rôle à expliquer
    {
        Pays p = null;
        for(int i=0; i<this.lesPays.length; i++)
            if (this.lesPays[i].getNom().equals(nomP))          p=this.lesPays[i];
        return p;
    }

    private void BajouterActionPerformed(java.awt.event.ActionEvent evt) { // à compléter
        if ( !Nom.getText().equals("") && !Prenom.getText().equals("")) {
            String poste = ...
            String nomPays= ...
            Pays pays = ...
            ...
        }
    }

    ...

    private void ParcourirActionPerformed(java.awt.event.ActionEvent evt) { // à expliquer
        JFileChooser jf = new JFileChooser();
        if (jf.showOpenDialog (this)==JFileChooser.APPROVE_OPTION)
        {
            this.img = new Imagelcon(getClass().getResource(jf.getSelectedFile().getName()));
            Photo.setText(jf.getSelectedFile().getName());
        }
    }
}
```