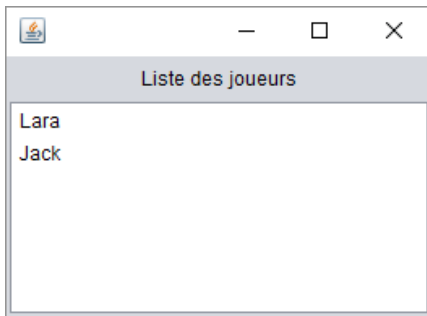


Exercice 1 : Interface graphique et boîte de dialogue - Gestion des Joueurs - Visualisation

On souhaite implémenter l'application nommée « GestionJoueurs » présentée au CM4 qui comporte une fenêtre principale dont l'interface est rappelée ci-dessous. Cette interface comporte uniquement un message de titre « Liste des joueurs » et une liste de noms de joueurs qui ont été directement insérés dans le code du programme. Cette liste est de type « JList » et est nommée « ListeJoueurs ». Pour rappel les informations concernant les joueurs ont été simplifiées, et sont : son pseudo, sa couleur, son solde et une photo (conservée sous forme d'une image icône de type « ImageIcon »).

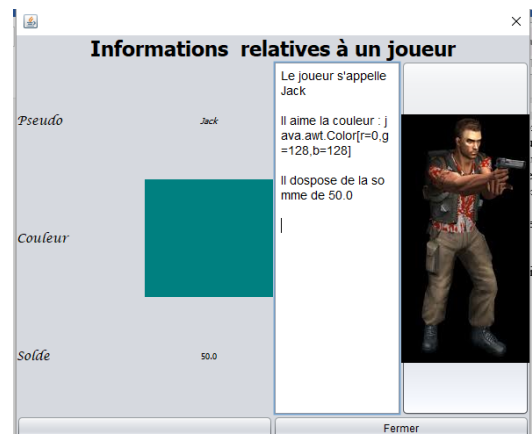
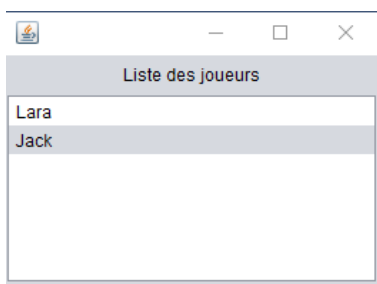


La classe « Joueur » donnée en annexe » représente un joueur. Cette classe est identique à celle donnée en CM et déjà utilisée dans TP précédent.

Cette classe comporte deux constructeurs.

Un constructeur par défaut, et un constructeur avec un paramètre qui est le pseudo du joueur. Les deux constructeurs initialisent la photo du joueur avec une image icône donnée par défaut (construite à partir du fichier image joueurDefault.png).

Lors de la sélection d'un nom de joueur dans la liste, une boîte de dialogue « VisuJoueurDlg » est ouverte qui décrit le détail de ce joueur et affiche sa photo.



Les informations à gérer dans l'application principale nommée « GestionJoueurs » sont uniquement une liste de joueurs. Un attribut de type « ArrayList<Joueur> » nommé « lst », sera utilisé pour gérer cette liste.

Le constructeur de cette classe assure :

- la construction de l'interface (appel de la méthode « initComponents() »)
- la création et l'ajout de 2 joueurs dans la liste « lst » (appel d'une méthode nommée « initJoueurs »)
- le remplissage de la liste (JList nommée ListeJoueurs) avec les pseudos des joueurs (appel d'une méthode nommée « initListeJoueurs »)

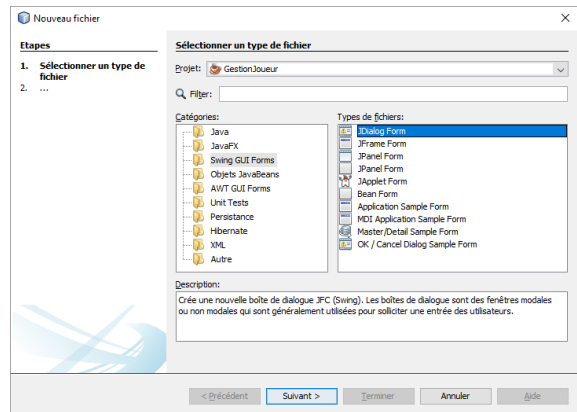
Le seul événement géré par cette classe est la sélection du nom d'un joueur qui ouvre une boîte de dialogue descriptive du joueur comme le montre l'interface ci-dessus à droite.

La réalisation de cette application nécessite l'implémentation de 3 classes : la classe principale « GestionJoueurs », la classe « Joueur » et une classe boîte de dialogue « VisuJoueurDlg » qui est une sorte de « JDialog ».

1. Créer un nouveau projet nommé « GestionJoueurs », et ajouter dans ce projet une nouvelle classe de type « JFrame » également nommée « GestionJoueurs ».
2. Ajouter dans ce projet une classe java « simple » nommée « Joueur ».

3. Ajouter également dans le projet une classe de type boîte de dialogue, en suivant les étapes ci-dessous :

- a. Clic droit sur le projet -> « Nouveau » et choisir « JDialog Form » si cette option est disponible, ou sinon, choisir l'option « Autre » et dans la boîte de dialogue qui s'ouvre, choisir « Swing GUI Form » à gauche et « JDialog Form » à droite.
- b. Puis choisir le nom de la boîte à créer, qui sera nommée « VisuJoueurDlg ».



4. Copier également dans le répertoire « src » de votre projet, les images disponibles en ligne avec le sujet de ce TP (à savoir les fichiers « joueurDefault.png », « jack.png », « lara.jpg »). Ce sont les mêmes images que celles récupérées lors du précédent TP

5. Classe « Joueur »

- a. En utilisant l'annexe, compléter le code de la classe joueur de votre application. Il y a déjà tout dedans ???

6. Classe « GestionJoueurs » - Classe principale

- a. En utilisant l'EDI, construire l'interface de la classe principale (qui comporte une « JList » nommée « ListeJoueurs »). Le mode de sélection dans cette liste est restreint à une simple sélection, pour cela il faut fixer la valeur de la propriété « selectionMode » de cette liste à « SINGLE ».
- b. Ajouter un attribut dans la classe : `private ArrayList<Joueur> lst;`
- c. Compléter son constructeur comme décrit ci-dessous, et commenter votre code en expliquant chaque ligne de ce constructeur (se référer au CM2 pour la gestion du modèle de la liste)

```
public GestionJoueurs() {
    initComponents();
    lst=new ArrayList<Joueur>();
    DefaultListModel mod = new DefaultListModel();
    ListeJoueurs.setModel(mod);
    initJoueurs();
    initListeJoueurs();
}
```

- d. Implémenter la méthode « initJoueurs » qui permet d'ajouter deux joueurs : « Lara », avec un solde de 100 et la couleur RGB (255, 215,0) et une image lara.jpg et « Jack », avec une couleur RGB(0,128,128), un solde de 50, et une image « jack.png ».

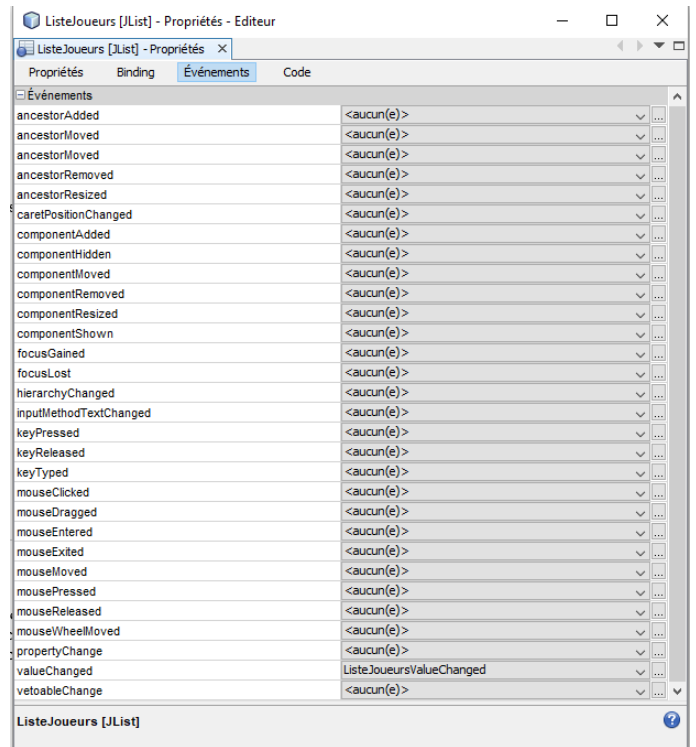
On rappelle que si l'image est disponible dans le répertoire « src » du projet, il est possible d'écrire l'instruction suivante pour affecter cette image à la photo du joueur (si j est de type Joueur)

```
j.setPhoto(new ImageIcon(getClass().getResource("/lara.jpg")));
```

- e. Implémenter la méthode de remplissage de la liste (JList) avec la liste des pseudos des joueurs.

f. La sélection d'un nom de joueur dans la liste entraîne l'ouverture d'une boîte de dialogue. Le gestionnaire d'évènement associé à cette sélection est décrit ci-dessous.

- i. Pour mettre en place ce gestionnaire, il faut sélectionner la JList dans la fenêtre de « Design », puis cliquer sur le bouton droit de la souris, choisir « Events » puis « ListSelection » et « valueChanged ». Le nom du gestionnaire proposé, est « ListeJoueursValueChanged » (ou MouseClicked)
- ii. Le code du gestionnaire peut alors être complété.
- iii. Commenter chaque ligne de ce code.



```
private void ListeJoueursValueChanged(javax.swing.event.ListSelectionEvent evt) {
    int ind= ListeJoueurs.getSelectedIndex();
    Joueur j= lst.get(ind);

    // appel de la boîte de dialogue de visualisation du joueur, le joueur à décrire dans la boîte de dialogue est passé
    // en 3ème paramètre du constructeur de la boîte de dialogue, il s'agit ici de l'objet j (de type Joueur)
    VisuJoueurDlg diag= new VisuJoueurDlg(this, true, j);

    diag.setSize(600, 500);
    diag.setLocation(100,100);
    diag.setVisible(true);
}
```

7. Classe « VisuJoueurDlg ».

Cette classe sert uniquement à visualiser les informations concernant un joueur. Les informations à gérer dans cette boîte de dialogue sont donc celles d'un joueur. L'adresse de ce joueur est récupérée en paramètre par le constructeur de la classe. C'est le constructeur de la classe qui est chargé d'afficher toutes les informations concernant le joueur. Le seul évènement géré est la fermeture de la boîte avec le clic sur le bouton « Fermer ».

- a. Construire l'interface de cette boîte de dialogue en utilisant l'EDI. La photo du joueur est affichée sur un bouton (JButton nommé Photo), les informations textuelles, sont dans une zone d'édition de type JTextArea nommée Edition (elle-même placée dans un JScrollPane). La couleur favorite du joueur est donnée comme couleur de fond d'un JPanel. Les autres éléments sont des JLabel ou (des JTextField non éditables éventuellement).
- b. Ajouter un attribut pour gérer le joueur : private Joueur lejoueur;
- c. Compléter le constructeur selon le code ci-dessous, en commentant les deux dernières instructions de ce code.

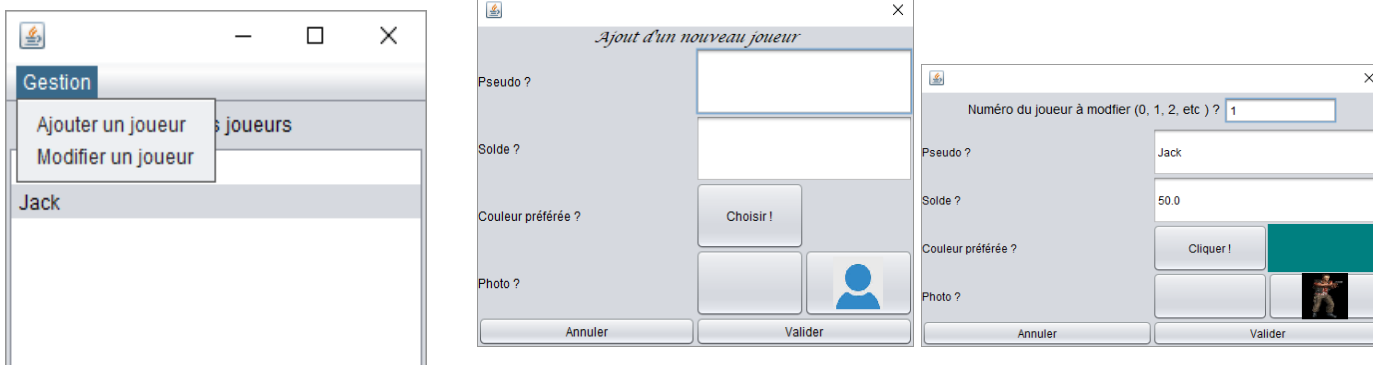
```
public VisuJoueurDlg(java.awt.Frame parent, boolean modal, Joueur j) {
    super(parent, modal);
    initComponents();
    this.lejoueur=j;
    initInfos();
}
```

- d. Donner le code de la méthode « initInfos » qui :
 - Affiche le pseudo et le solde du joueur dans les zones respectives
 - Colorie le panneau avec la couleur du joueur
 - Affiche les informations textuelles dans la zone d'édition,
 - Affiche la photo du joueur sur le bouton nommé « Photo » avec l'instruction Photo.setIcon(lejoueur.getPhoto());
- e. Implémenter le gestionnaire associé au clic sur le bouton « Fermer » qui ferme la boîte de dialogue. Les instructions : this.setVisible(false); this.dispose(); sont nécessaires pour fermer la fenêtre.

Exercice 2 : Interface graphique et boîte de dialogue - Gestion des Joueurs - Ajout

On souhaite compléter cette application afin de permettre l'ajout de nouveaux joueurs et la modification des joueurs existants, un menu est ajouté avec une option « Gestion » et deux sous-options « Ajouter un joueur » et « Modifier un joueur » comme décrit sur l'interface ci-dessous. L'ajout d'un joueur se fait en utilisant une boîte de dialogue qui permet la saisie du joueur, et la modification d'un joueur se fait avec une boîte de dialogue similaire mais pré-remplie avec les informations du joueur à modifier lorsqu'il a été sélectionné. Dans un 1^{er} temps, une photo par défaut est affectée au joueur et n'est pas modifiable.

La description complète de cette application est donnée dans le CM4. Le code complet de l'ajout d'un joueur a été écrit en CM.

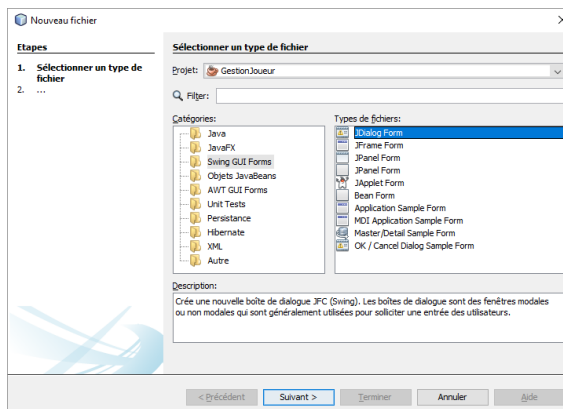


8. Ajouter dans le projet « GestionJoueurs » une classe de type boîte de dialogue, en suivant les étapes ci-dessous :

- Clic droit sur le projet -> « Nouveau » et choisir « JDialog Form » si cette option est disponible, ou sinon, choisir l'option « Autre » et dans la boîte de dialogue qui s'ouvre, choisir « Swing GUI Form » à gauche et « JDialog Form » à droite.
- Puis choisir le nom de la boîte à créer, qui sera nommée « SaisieJoueurDlg ».

9. Compléter cette classe « SaisieJoueurDlg » en utilisant vos documents de CM (CM4).

10. Ajouter dans la classe « GestionJoueurs » un évènement de type Action, pour gérer le clic sur la sous-option « Ajouter un joueur » de l'option « Gestion » du menu, et compléter le gestionnaire associé en vous aidant du code ci-dessous et en le complétant.



```
private void AjouterActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    SaisieJoueurDlg diag= new SaisieJoueurDlg (this, true); // déclaration et création de la boîte de dialogue
    diag.setVisible(true); // affichage de la boîte de dialogue
    if (diag.getOk() == true ) // test de l'indicateur de fermeture
    { Joueur j = diag.getJoueur(); // récupération du joueur qui a été saisi dans la boîte
```

```
    // A COMPLETER
    // ajout du joueur dans la liste des joueurs
    ...
    // ajout de son pseudo dans la liste (JList ListeJoueurs)
    ...
}
```