CNAM

M-NFA032 Algorithmique et programmation : bibliothèques et patterns

Neuvième partie

Travail à effectuer

Rappels

Quelques éléments importants avant de commencer le fil rouge :

- à terme, votre programme comportera une interface graphique ; cette interface ne vous permettra pas de lire au clavier ni d'écrire à l'écran ; donc, ne pas mettre d'ordre de lecture ou d'écriture dans vos classes ; si vous souhaitez valider vos classes, prévoyez alors une petite bibliothèque temporaire pour les entrées-sorties.
- Dans vos classes, mettez systématiquement vos champs private ou, à la limite, protected ; vous pourrez ensuite donner des accès à vos variables d'instances via des getters et des setters.

Thème de la semaine

Cette semaine, nous allons avancer le formulaire d'identification.

Exercice 1 : Quitter l'application

Nous allons créer une action « QuitterAction » qui sera utilisée à plusieurs endroits dans l'application.

1. Dans le package ui.commun, créez une classe « QuitterAction » qui étend « AbstractAction ».
2. Donnez-lui « Quitter » comme nom par défaut.
3. Ajoutez un constructeur qui prend en paramètre le nom de l'action. Ceci sera utile pour le 3ème exercice de cette feuille.
4. Comme action, testez si les données ont été modifiées ou non et, selon le cas, demandez confirmation à l'utilisateur. Selon le résultat, quittez l'application avec System.exit(0) ; ou ignorez l'action.

Exercice 2 : Observateur de fenêtres

Pour surveiller un éventuel clic de l'utilisateur sur le bouton de fermeture de fenêtre, nous allons créer un « écouteur de fenêtre ».

1. Dans ui.commun, créez une classe « EcouteurFermeture » héritière de « WindowAdapter ».
2. Surchargez la méthode WindowClosing comme dans l'exemple du cours en l'adaptant au besoin afin de simuler l'action définie dans l'exercice précédent.
3. Dans les constructeurs des classes « LoginFrame » et « MainFrame » :
    1. lors de l'appel à setDefaultCloseOperation, remplacez « EXIT_ON_CLOSE » par « DO_NOTHING_ON_CLOSE ». C'est nous qui allons gérer la fenêtre.
    2. Ajoutez à la suite de la ligne setDefaultCloseOperation(...) ; la ligne suivante afin que l'observateur que nous venons de définir surveille la fermeture de la fenêtre.

addWindowListener(new EcouteurFermeture());

Vous pourrez tester le comportement du bouton X de la fenêtre d'identification.

Exercice 3 : Bouton « Annuler » de la fenêtre d'identification

Le rôle de ce bouton est d'arrêter le programme. Nous allons le faire en créant une class héritière de « QuitterAction ».

1. Créez une action « LoginAnnulerAction » dans ui.loginframe qui étend QuitterAction.
2. Ajoutez un constructeur qui appelle celui qui a été créé à la question 1.3. Le paramètre sera le texte du bouton.
3. Modifiez le bouton « Annuler » du formulaire d'identification pour qu'un appui dessus déclenche l'action que vous venez de créer.

Testez votre nouveau bouton...

Exercice 4 : Bouton Connexion

1. Créez une action dans ui.loginframe.
2. Comme action, appelez simplement une méthode statique et sans paramètre, nommée « validateLogin » de la classe « Systeme ».
3. Créez cette méthode dans la classe Systeme.
   1. Vérifiez que lf ne vaut pas null, sinon, levez une RuntimeException spécifique pour signaler que la méthode a été appelée en dehors du formulaire d'identification.
   2. Déléguez à lf la vérification des identifiants.
4. Continuez à déléguer jusqu'au panneau qui a créé les champs de saisie du formulaire d'identification.
5. Là, récupérez les données saisies par l'utilisateur.
   - l'email à l'aide de la méthode getText() sur le JTextField,
   - le mot de passe à l'aide de getPassword()[1] sur le JPasswordField() en pensant à convertir le tableau de char obtenu en une String.
6. Cherchez l'utilisateur dans le carnet de clientèle et vérifiez que le mot de passe correspond. En cas d'erreur, email inconnu ou mot de passe erroné, affichez un message d'erreur dans la zone prévue à cet effet dans le formulaire.
7. Si les données d'identification sont correctes, connectez l'utilisateur à l'aide de la méthode setUtilisateur de la classe Systeme.
8. Associez cette nouvelle action au bouton OK du formulaire d'identification.

Vous pouvez tester cette nouvelle action. La fenêtre principale devrait s'ouvrir avec un login correct (celui que vous avez défini dans la méthode initSysteme lors de l'exercice 7.3

Exercice 5 : Ouverture du formulaire de création d'un compte

Nous allons procéder d'une manière un peu différente pour enregistrer un nouvel utilisateur. Les actions associées aux boutons seront des instances d'une classe encapsulée dans le panneau de formulaire.

Pour cet exercice et le suivant, nous travaillerons dans le package « ui.client »

1. Créez un JPanel « UtilisateurBodyPane » représentant un formulaire en vue de créer un nouveau compte utilisateur.
   - Le mot de passe devra être demandé deux fois pour confirmation !
   - Prévoyez une zone pour afficher les erreurs.

1 getPassword() retourne un char[], vous pouvez le convertir en String avec un new String (xxx.getPassword()) où xxx désigne la variable qui désigne le JPasswordField.

- Le constructeur recevra en paramètre un JDialog qui sera mémorisé dans une variable d'instance « dialog ». Nous en verrons l'utilité plus tard.

2. Créez dans la classe « UtilisateurBodyPane », une sous-classe privée « AnnulerAction » étendant « AbstractAction ».
   1. Nommez cette action « Annuler » pour que ce texte apparaisse dans le bouton.
   2. L'action à exécuter consistera uniquement à fermer le formulaire de saisie avec l'instruction :
      ```
      dialog.dispose();
      ```
3. Comme précédemment, dans « UtilisateurBodyPane », créez une sous-classe « EnregistrerAction » étendant elle aussi « AbstractAction ».
   1. Mettez « Enregistrer » comme nom pour l'action.
   2. Pour l'action à proprement parlé :
      1. créez une chaîne de caractères erreur initialement vide ;
      2. vérifiez que tous les champs sont renseignés ; si l'un d'entre eux ne l'est pas, ajoutez un message à erreur le signalant et se terminant par la balise html
 pour passer à la ligne ;
      3. Vérifiez que l'email n'est pas déjà utilisé dans le carnet de clientèle ;
      4. Vérifiez que les deux mots de passe sont identiques ;
      5. à ce point, si erreur n'est pas vide (sa longueur est différente de 0), mettez son contenu entouré des balises <html> et </html> dans la zone d'affichage des erreurs que cous avez construite précédemment puis quittez la méthode avec un simple return.
      6. Sinon, créez un nouveau client avec les données introduites par l'utilisateur.
      7. Ajoutez-le au carnet de clientèle.
      8. Sauvegardez les données à l'aide du gestionnaire de sauvegarde.
      9. Fermez la boîte de dialogue comme pour l'action annuler.
      10. Appelez la méthode setUtilisateur de Systeme avec le client.
4. Ajoutez au corps de « UtilisateurBodyPane » une méthode newEnregistrerAction retournant une « AbstractAction ». Elle devra retourner une nouvelle instance de EnregistrerAction.
5. Faites de même pour « AnnulerAction ».

Exercice 6 : Finalisation

1. Créez une classe étendant JPanel avec trois parties :
   1. un titre
   2. le formulaire créé dans l'exercice précédant (prendre en paramètre du constructeur le JDialog qui sera transmis au formulaire.
   3. une zone de deux boutons. Cette zone prendra l'instance de formulaire crée lors de l'étape précédente en paramètre et l'utilisera pour définir les actions des deux boutons : enregistrer et annuler.
2. Créez un JDialog qui contiendra le JPanel défini ci-dessus. Pensez à transmettre l'instance en cours au JPanel précédent afin que l'action « EnregistrerAction » puisse fermer ce JDialog.
3. Créez une action qui ouvrira le JDialog ci-dessus
4. Affectez cette action au deuxième bouton du formulaire d'identification.
5. Testez.