Dans cette fiche d'exercices, il est demandé de faire l'exercice 1 puis d'en choisir un des trois autres. Quelques rappels sur les boîtes de dialogues sont disponibles à la fin de la fiche.

1 Les exercices

Exercice 1.
Reprendre le fichier `Palindrome.java` envoyé dans vos espaces et par mail et le modifier en utilisant :
– une boîte d'entrée pour la saisie du texte ;
– une boîte de message pour la réponse.

Exercice 2.
Créer un jeu où l'ordinateur choisit un nombre entre 1 et 100 et l'utilisateur doit le chercher en proposant des valeurs. À chaque valeur proposée (dans une boîte de dialogue), l'ordinateur répond « plus grand », « plus petit » ou « gagné ! » (dans une boîte de dialogue...).

Exercice 3.
Créer un jeu qui demande de penser à un nombre à l'utilisateur puis cherche ce nombre en proposant des valeurs et en demandant si le nombre cherché est plus grand, plus petit ou égal. (Ici c'est l'ordinateur qui « joue » ; l'utilisateur répond à ses questions en cliquant sur les boutons d'une boîte de dialogue).

Exercice 4.
Créer un programme qui permet de tester les connaissances d'un candidat sous la forme d'un qcm. Cahier des charges :
– pour chaque question on propose trois réponse dont une seule est juste ;
– les questions, propositions et réponses seront stockés (pour le moment [1]) dans un tableau ;
– les réponses doivent se faire en cliquant sur un bouton ;
– un message doit afficher à la fin le score du candidat (avec ou sans pénalisation des mauvaises réponses).

2 Quelques rappels

Penser à indiquer : `import javax.swing.JOptionPane;` dans le préambule (mais Eclipse vous le rappelle...)

2.1 Boîte de confirmation

Une boîte de confirmation demande un « clic » à l'utilisateur (oui/non, OK/Annuler/Ignorer, ...). La réponse est un `int`. On la déclare ainsi :

```
int reponse;
reponse = JOptionPane.showConfirmDialog(null,// position
     "Vous avez tout compris ?", // la question
     "Question",// intitulé de la boîte
     JOptionPane.YES_NO_OPTION,// type de boutons
     JOptionPane.INFORMATION_MESSAGE);// icône
```

En sortie, `reponse` vaut 0 si on clique sur oui et 1 si on clique sur non.

1. Un projet intéressant serait de créer un logiciel qui permet au professeur de construire un ou plusieurs qcm et un autre qui permet à l'élève de le passer.

2.2 Boîte d'entrée de texte

Une boîte d'entrée permet à l'utilisateur du programme de saisir du texte (à vous de le convertir si vous souhaitez un autre format de données...) :

```
String reponse;
reponse = JOptionPane.showInputDialog(null,
    "Menteur ! Résumez vos connaissances ici :",
    "Saisir du texte",
    JOptionPane.QUESTION_MESSAGE);
```

La variable `reponse` contient :
- le texte saisi si on a cliqué sur OK ;
- `null` si on a cliqué sur Annuler.

2.3 Boîte de message

Parfois, on souhaite juste afficher une réponse ou une information :

```
JOptionPane.showMessageDialog(null,
    "Il va falloir travailler plus",
    "Conseil important !",
    JOptionPane.WARNING_MESSAGE);
```

2.4 Personnalisations

Dans Eclipse, chaque fois que vous saisissez « `JOptionPane.` » une info-bulle vous propose plusieurs suggestions, à vous d'essayer d'autres types de « `Pane` » (ou panneaux) et de voir ce que cela donne... Un exemple de fenêtre avec quatre boutons personnalisés (pour d'autres exemples, internet est votre ami) :

```
JOptionPane d = new JOptionPane();
// les textes figurant sur les boutons
String lesTextes[]={ "bonjour",    "au revoir",
                     "bonne nuit", "bonne année"};
int  retour  = // indice du bouton qui a été cliqué ou CLOSED_OPTION
     d.showOptionDialog(getFrame(), "le message", "le titre",
                        JOptionPane.YES_NO_OPTION,
             JOptionPane.INFORMATION_MESSAGE,
                        null,           // pas d icone
                        lesTextes,      // les textes de boutons
                        lesTextes[0]);  // le bouton par défaut
      if( retour!=JOptionPane.CLOSED_OPTION);// un bouton cliqué
      else ;                                 // pas de bouton cliqué
```

Corrigés des exercices