

Travaux pratiques d'informatique N° 21

Le but principal de cette séance est de vous permettre d'améliorer vos connaissances concernant la programmation graphique et la gestion des événements (y compris l'utilisation des fichiers multimédia, la réalisation des animations et le contrôle des interfaces graphiques par l'intermédiaire des événements liés à la souris et au clavier).

1. Reprendre et modifier la calculatrice graphique réalisée dans les Travaux pratiques d'informatique N° 19 en respectant les consignes suivantes :

- les calculs sont affichés à la fois dans la fenêtre console et dans une zone d'affichage située au-dessus des touches de la calculatrice, comme dans la figure ci-dessous ;



- la zone d'affichage utilise une police **Arial** 24 grasse et les opérandes et les opérateurs apparaissent d'abord en bleu et finalement, après avoir effectué chaque calcul, en rouge ;
- afin d'introduire des données, l'utilisateur peut employer la souris ou le clavier (les seules touches "sensibles" du clavier étant celles équivalentes aux touches de la calculatrice).

Indications :

- modifiez le design de la GUI obtenue dans les Travaux pratiques d'informatique N° 19 en ajoutant au conteneur de premier niveau **JFrame** une étiquette **JLabel** pour la zone d'affichage et un panneau **JPanel** pour la zone de touches (l'opérateur de multiplication étant représenté par *) ;
- modifiez la méthode **ActionPerformed** de sorte qu'elle appelle une nouvelle méthode nommée **calculerAfficher** qui fait les calculs et affiche les résultats dans la fenêtre console et dans la zone d'affichage de la calculatrice ;
- ajoutez un écouteur de type **KeyListener** et définissez d'une manière appropriée la méthode **keyPressed** (qui fera appel à la nouvelle méthode **calculerAfficher**).

2. Concevoir une application graphique autonome qui permet à l'utilisateur d'interagir avec une interface graphique (GUI) qui peut être vue comme le clavier d'un piano virtuel simplifié (qui a seulement trois touches).

Au début de l'exécution de l'application, une interface graphique comme celle présentée dans la figure ci-dessous doit apparaître au coin supérieur gauche de l'écran. Les textes associés aux 5 boutons sont écrits en utilisant une police de la famille **Arial**, grasse et de taille 20 pixels. Les images associées aux 3 "touches" du clavier correspondent aux fichiers **im1.jpg**, **im2.jpg** et **im3.jpg** disponibles pour téléchargement sur le Bureau virtuel du CMS.



Normalement, quand l'utilisateur appuie (avec la souris) sur un des trois boutons dont les textes associés sont **Do**, **Mi** et **Sol** ou quand l'utilisateur appuie sur une des trois touches **c**, **e** et **g** du clavier, un fichier son (i.e. un fichier audio) approprié est joué (une seule fois), à savoir le fichier **do.wav** pour un clic sur le bouton **Do** ou sur la touche **c**, le fichier **mi.wav** pour un clic sur le bouton **Mi** ou sur la touche **e**, et le fichier **sol.wav** pour un clic sur le bouton **Sol** ou sur la touche **g**. Les 3 fichiers audio mentionnés sont disponibles pour le téléchargement sur le Bureau virtuel du CMS.

Quand l'utilisateur appuie (avec la souris) sur le bouton dont le texte associé est **PLAY**, l'interface graphique devient une sorte de jukebox et :

- les boutons **Do**, **Mi** et **Sol** seront appuyés :
 - o successivement (un après l'autre, à **1 seconde** d'intervalle, en commençant avec le bouton **Do**) ;

- automatiquement (par le programme, sans l'intervention de l'utilisateur par l'intermédiaire de la souris ou du clavier) ;
- en boucle (i.e. après le bouton **Sol** on continuera avec le bouton **Do** et ainsi de suite) ;
- de plus, chaque fois qu'un des trois boutons **Do**, **Mi** et **Sol** sera appuyé automatiquement, le fichier son correspondant sera joué ;
- pendant ce temps, des clics avec la souris sur les boutons **Do**, **Mi** ou **Sol**, ainsi que l'appui sur des touches du clavier (y compris sur les touches **c**, **e** et **g**) resteront sans aucun effet ;
- les sons seront joués en boucle jusqu'à ce que le bouton dont le texte associé est **STOP** soit appuyé par l'utilisateur, à l'aide de la souris (par contre, si le bouton **STOP** est appuyé à d'autres moments, cette action restera sans effet).

Indications :

- écrivez une classe graphique englobante nommée **Piano**, qui hérite d'un container swing de premier niveau et qui définit tous les champs (privés) dont elle a besoin ;
- dans la classe englobante **Piano**, définissez une classe interne écouteur (nommée **EcouteurPiano**) qui implémente les interfaces nécessaires et (re)définit les gestionnaires d'événements appropriés afin de rendre les 5 boutons réactifs (à la fois aux clics de la souris et à certaines touches du clavier) et de permettre "l'animation auditive" du genre jukebox ;
- dans la classe englobante **Piano**, définissez une classe graphique interne (nommée **Clavier**) qui correspond au clavier du piano virtuel ; cette classe interne hérite d'un container intermédiaire auquel vous ajoutez (dans le constructeur de la classe **Clavier**) 3 instances (créées pour l'occasion) de type bouton swing ainsi qu'une instance de la classe non graphique **javax.swing.Timer** (qui est le "moteur" du jukebox) ;
- dans le constructeur de la classe englobante **Piano**, réalisez principalement le design de la GUI montrée dans la figure de la page 2 ;
- écrivez la classe principale qui, dans la méthode **main**, crée une instance de la classe graphique englobante.

Remarque : le même effet que celui produit quand l'utilisateur clique avec la souris sur un bouton swing peut être obtenu à l'aide du code en appelant la méthode d'instance **doClick()** (sans argument) pour le bouton respectif.