

PARTIEL 2

Seuls sont autorisés, à titre de documents, les listings comprenant explicitement le nom de l'étudiant (et celui-ci seulement) et les notes manuscrites.

Exercice 1.- (File d'attente)

Une **file d'attente** est une structure de donnée dynamique qui stocke les données et les traite dans l'ordre d'arrivée (premier arrivé, premier servi). Seules deux opérations sont permises pour accéder à celle-ci : **placer** ajoute un item en queue de la file d'attente et **enlever** retire un item en tête de la file d'attente.

- 1°) Définir les classes **item** des éléments d'une file d'attente d'entiers naturels et **queue** de file d'attente elle-même en C++.

[Les données seront un pointeur d'item pour indiquer la tête et un pointeur d'item pour indiquer la queue. Le constructeur initialise les deux pointeurs à la valeur nulle. La méthode booléenne **vide()** teste si la file est vide. La méthode **placer(entier)** ajoute un élément en tête. La méthode **enlever()** renvoie l'entier en tête si la file d'attente n'est pas vide et -1 sinon.]

- 2°) Écrire un programme C++ qui permet de saisir un certain nombre d'entiers strictement positifs, la valeur sentinelle étant -1, puis un entier naturel n . Le programme affichera les entiers saisis depuis le n -ième jusqu'à la fin (le premier entier entré a pour numéro 0).

Les entiers saisis seront sauvegardés dans une file d'attente.

Exercice 2.- (Événements du clavier)

Écrire une application Java qui fait apparaître un cadre dans lequel sont indiquées les touches sur lesquelles on a appuyé et les touches qu'on a relâché. Un exemple de session devrait être :

```
Touche frappée      :  a
Touche frappée      :  L
Touche frappée      :  z
Touche relâchée     :  L
Touche relâchée     :  z
Touche relâchée     :  a
```

avec un déroulement sur dix lignes.

[En fait la répétition automatique des touches du clavier fait que tout va très vite. Il faut annihiler celle-ci pour obtenir une session intéressante.]

Exercice 3.- (AudioClip)

Écrire une applet faisant apparaître trois boutons : “jouer”, “arrêter” et “jouer en continu”. Lorsqu'on appuie sur l'un des boutons extrêmes, un morceau musical (disons “parrain.mid”) est exécuté, en bouclant sur lui-même dans le deuxième cas. Lorsqu'on appuie sur le bouton du milieu, le morceau musical est interrompu.

[Ne pas oublier le fichier html.]

DOCUMENTATION

1 Les événements claviers

Les événements du clavier sont traités grâce à l'interface :

`KeyListener`

dont il faut surcharger les trois méthodes, sans exception :

```
public void keyPressed(KeyEvent )
public void keyReleased(KeyEvent )
public void keyTyped(KeyEvent )
```

qui réagissent respectivement à la pression d'une touche, au relâchement d'une touche et à la pression d'une touche qui n'est pas une touche d'action.

La méthode :

```
addKeyListener(ObjetEcouteEvenement)
```

de `ObjetSourceEvenement` permet de se mettre à l'écoute des événements du clavier.

La méthode :

```
char getKeyChar()
```

de la classe `KeyEvent` donne la valeur Unicode du caractère tapé au clavier.

2 Les morceaux musicaux

Les morceaux musicaux ne peuvent être joués, pour l'instant, que dans des applets ou avec un complément Java appelé `Java Media Framework`. Ce sont des objets de la classe :

`AudioClip`

On récupère un morceau musical grâce à la méthode :

```
AudioClip getAudioClip(getDocumentBase(), String)
```

de la classe `Applet`, la chaîne de caractères étant le nom du fichier (`.au`, `.wav`, `.aif`, `.mid` ou `.rmi` uniquement ; pas de `mp3` pour les applets).

La classe `AudioClip` possède les trois méthodes :

```
play()
loop()
stop()
```

dont les noms sont parlants.