# TD/TP 1 – Rappels

## Exercices

### Exercice 1 : Classe Abstraite, Interface & Héritage

Pour l’exercice suivant les classes se situent dans les *packages* ***edu.iut.app*** et ***edu.iut.gui.listeners***

* Compléter l’interface ***IApplicationLogListener*** ayant une méthode ***void newMessage(String level, String message)***
* Compléter une interface ***IApplicationLog*** ayant les méthodes suivantes
* ***void setMessage(String message)***
* ***String getMessage()***
* ***void addListener(IApplicationLogListener listener)***
* ***IApplicationLogListener[] getpplicationLogListeners()***
* Compléter la classe ***AbstractApplicationLog*** implémentant l’interface ***IApplicationLog***
  + La méthode ***setMessage*** sera abstraite
  + Cette classe devra contenir un tableau de ***IApplicationLogListener*** (sous forme de collection) avec une méthode d’ajout et une méthode permettant d’obtenir le tableau/
* Compléter les classes ***ApplicationErrorLog, ApplicationWarningsLog, ApplicationInfoLog*** dérivant de ***AbstractApplicationLog*** 
  + Ces classes devront contenir un tableau de ***IApplicationLogListener*** (sous forme de collection)
  + La méthode ***setMessage*** doit appeler la méthode ***newMessage*** de chaque élément du tableau de ***IApplicationLogListener*** et conserver le message
* Compléter la boîte de dialogue abstraite ***AbstractApplicationMessageDialog*** implémentant l’interface ***IApplicationLogListener***
  + ajouter une méthode abstraite ***void showMessage( )***, cette méthode doit être accessible par les classe filles seulement.
  + la méthode ***void newMessage(String message)*** doit appeler la méthode ***showMessage( )***
* Compléter les 3 boites de dialogue dérivant de ***AbstractApplicationMessageDialog***:
  + ***ApplicationErrorMessageDialog ,*** affichant un message d’erreur
  + ***ApplicationWarningMessageDialog***, affichant un avertissement
  + ***ApplicationInfoMessageDialog***, affichant un message

### Exercice 2 : Collections

Créer une table de logs ***ApplicationLogs***

* La classe ***ApplicationLogs*** doit dériver d’une Collection représentant un tableau de ***IApplicationLog***
* Implémenter les fonctions suivantes permettant de catégoriser les erreurs
  + ***ArrayList<IApplicationLog> getErrors(),***
  + ***ArrayList<IApplicationLog> getWarnings(),***
  + ***ArrayList<IApplicationLog> getInfos()***

### Exercice 3 : GUI – Visualisation de Widgets sous forme de table : implémentation partielle d’un agenda

* Implémenter 3 vues (Mois / Semaine / Jour) : classes ***EventPanel, MonthPanel, WeekPanel, DayPanel***
* Trouver un *Layout* permettant de superposer ces trois vues : classe ***SchedulerFrame***
* Créer une ***Factory*** (classe ***AgendaPanelFactory***)permettant de gérer les trois classes ***MonthPanel, WeekPanel, DayPanel***
* Créer une Panel permettant de gérer les vues ***MonthPanel, WeekPanel, DayPanel :*** en cliquant sur un bouton ‘***next***’ nous devons pouvoir voyager entre les vues

## Aide

* Operateur ***instanceof***
* ***JOptionPane***: <https://openclassrooms.com/courses/apprenez-a-programmer-en-java/les-menus-et-boites-de-dialogue>

## Rapport

### Description des travaux

Nous avons corrigé et complété le code donné, notamment en :

* implémentant les méthodes créant des messages d'erreur en ouvrant de nouvelles fenêtre selon le niveau de gravité de l'erreur
* définissant les méthodes des interfaces et en ajoutant les méthodes manquantes aux classes
* créant les constructeurs manquants
* implémentant la classe abstraite AbstractApplicationLog
* utilisant des collections ArrayList.

### Difficultés rencontrées

Manier un grand nombre de classes plus ou moins reliées par des interfaces, classes et méthodes abstraites.