

Interfaces and Inheritance

Exercice Avec une liste

On reprend les classes `Box` et `TextBox` du premier TD. Si on voulait réaliser une interface graphique, il faudrait gérer une structure qui rassemble des instances de `Box`. Implementez les méthodes suivantes (respectez bien les signatures) dans une classe `ListGI` (Comme pour `ArrayList` ou `LinkedList`, ici, je choisis le nom `ListGI` pour dire que c'est une Graphical Interface basée sur une liste).

1. **public void** `addBox(int tl_x, int tl_y, int br_x, int br_y)` qui ajoute une `Box` dont les coordonnées du coin en haut à gauche est `(tl_x, tl_y)` et les coordonnées du point en bas à droite est `(br_x, br_y)`.
2. **public void** `display(int n)` qui affiche les informations de la box dont l'identifiant est passé en paramètre. Si l'identifiant correspond à un `Box` qui n'existe pas, écrivez un message.
3. **public void** `removeBox(int n)` qui enlève la `Box` dont l'identifiant est `n`.
4. **public void** `listAllTopToBottom()` qui affiche toutes les `Box` par ordre de l'ordonnée du coin en haut à gauche.
5. **public void** `listAll()` qui affiche toutes les `Box` par ordre des identifiants.

N.B.: on pourrait reprendre l'exercice en utilisant une `Map` et non une liste.

Exercice Un petit menu

Ecrivez un menu qui contient les options suivantes:

1. "taper 0 pour sortir du programme"
2. "taper 1 pour ajouter un `Box`"
3. "taper 2 pour ajouter un `TextBox`"
4. "taper 3 pour afficher toutes les `Boxs` et `TextBoxs`"

Commencez par créer le menu, puis codez chaque fonctionnalité à l'aide de l'exercice précédent. Le menu doit être robuste aux fautes de frappe de l'utilisateur.