Héritage

On s'inspire de la création d'interface graphique. Nous n'allons pas en réaliser une aujourd'hui (c'est évidemment possible en Java), mais c'est juste pour avoir une image. Si on regarde une interface comme par exemple celle d'un gestionnaire d'emails, on voit beaucoup de rectangles : certains sont là pour donner un peu de couleurs, certains contiennent du texte, et certains contiennent du texte et sont des boutons. Donc on peut se dire que tout n'est que rectangle, qu'il y a des rectangles particuliers avec du texte, et qu'il y a des rectangles encore plus particuliers avec du texte et qui déclenche une action quand on clique dessus... Il semble que l'héritage soit bien utile ici ! On peut aussi voir quelques disques (eux aussi peuvent servir de boutons), et quelques lignes.

On va simplifier le problème. On travaille sur un écran de résolution 1920x1080, donc les coordonnées vont de 0 à 1079 en abscisse et 0 à 1019 en ordonnée. On va considérer que tous les rectangles sont horizontaux (pas de rectangles penchés).

1. Créez une classe `Box` qui sera définie par un identifiant de type `int` ainsi que par les coordonnées du coin en haut à gauche et du coin en bas à droite.
   - créez un constructeur qui prend en paramètres les coordonnées des deux coins et qui donne de manière automatique un unique identifiant.
   - créez une méthode `public int getId()` qui retourne l'identifiant de la `Box`
   - créer une méthode qui `String toString()` qui retourne une chaîne de caractères qui indique les caractéristique de la `Box`.
     Par exemple qui retourne la chaîne `"[Box id=13] (100, 200) - (300,100)"`.
   - écrire une méthode `main` qui crée deux `Box` et qui affichent leur caractéristiques.
2. Créez une classe `TextBox` qui n'est rien de plus qu'une `Box` à laquelle on ajoute du texte. Que proposez vous pour la méthode `toString`.
3. Qu'avez-vous choisi pour la visibilité de vos variables d'instance (par exemple pour les coordonnées des coins) ?

   ---

4. Mettez les variables d'instance en `private` (sans rien changer d'autres). Si vous avez une erreur de compilation, écrivez l'erreur ci-dessous

   ---

   Expliquez votre erreur et corrigez la.
5. Expliquez ci-dessous votre choix entre `private` et `protected`. Même si ce n'est pas la solution que vous privilégiez, on vous impose maintenant d'utiliser `private`. Modifiez votre code.

   ---

6. Implémentez une méthode `clone()` pour dupliquer un élément.
7. On va placer ces lignes de code dans la méthode `main`.

```
Box a = new Box(100, 200, 300,100);
Box b = a.clone();
System.out.print(a==b);
System.out.print("|"+a.equals(b));
```

   Modifiez si besoin votre code pour que l'exécution affiche `false|true` (ce ne sont pas les mêmes `Box`, mais deux `Box` avec les mêmes coordonnées sont égales.