TP N°6 ET 7 *Héritage et Interface*

Objectif

Travailler la notion d'héritage et interface

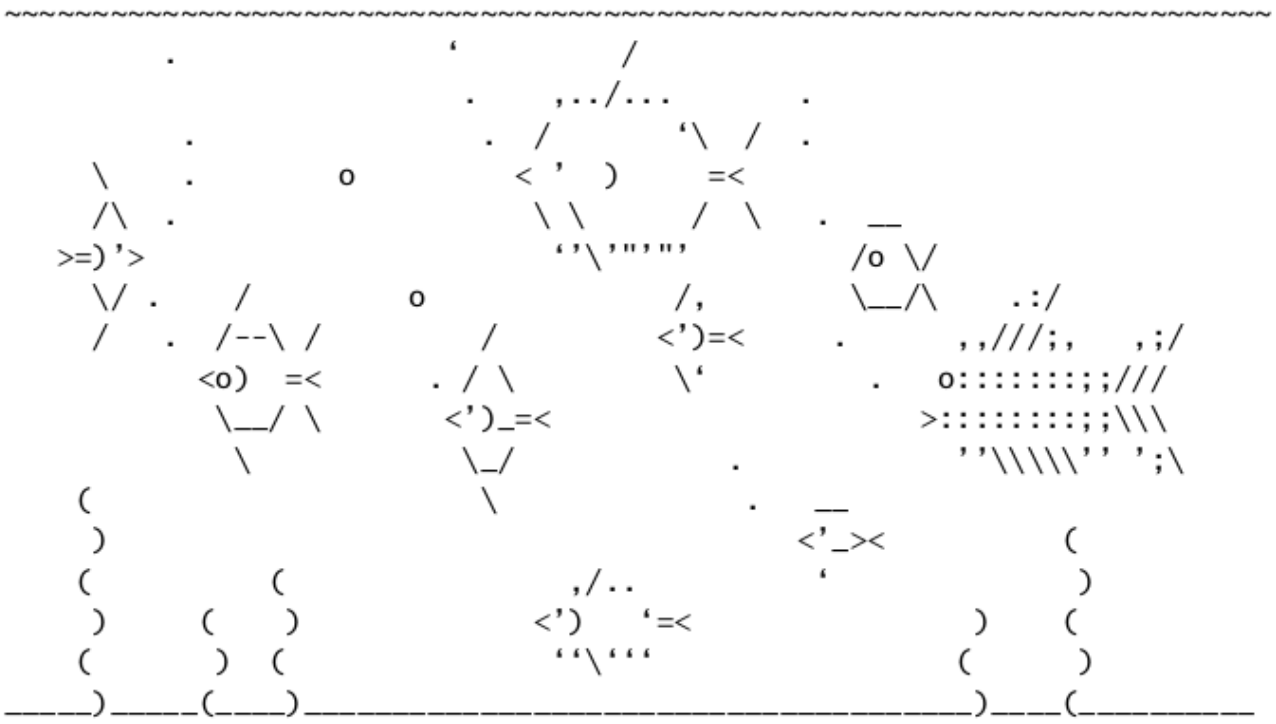
Exercice 1 *Refactorisation de code*

Factorise au mieux ce code pour éviter les codes redondants (tu peux ajouter de nouvelles classes)

```
public interface Outils {
  public void utiliser();
}

public class Marteau implements Outils {
  private String marque ;
  public Marteau (String m) { this.marque = marque; }
  public void taper() { System.out.print("taper");}
  public void utiliser() {
    System.out.print("mon marteau "+ marque +" permet de ");
    this.taper();
    System.out.println();
  }
}

public class Ciseau implements Outils {
  private String marque ;
  public Ciseau (String marque) { this.marque = marque;}
  public void couper() { System.out.print("couper");}
  public void utiliser() {
    System.out.print("mes ciseaux "+ marque +" permettent de ");
    this.couper();
    System.out.println();
  }
}

public class Secateur implements Outils {
  private String marque ;
  public Secateur (String marque) { this.marque = marque;}
  public void couper() { System.out.print("couper ");}
  public void utiliser() {
    System.out.print("mon sécateur "+ marque +" permet de ");
    this.couper();
    System.out.println();
  }
}

public class ExecutableOutils {
  public static void main( String [] args) {
    List<Outils> boiteAOutils = new ArrayList<>();
    boiteAOutils.add(new Marteau("Expert"));
    boiteAOutils.add(new Marteau("PowerGrip"));
    boiteAOutils.add(new Ciseau("HPC"));
    boiteAOutils.add(new Ciseau("SoftLine"));
    boiteAOutils.add(new Secateur("Laguiole"));
    for(Outils outil : boiteAOutils) {
      outil.utiliser();
    }
  }
}
```

Exercice 2 *Aquarium*

Dans cet exercice, on va réaliser un aquarium en ASCII art avec des poissons, des algues, des bulles, et tout ce qui vous fera plaisir, le tout animé.



Ce dessin a été repris de l'adresse : `http://www.retrojunkie.com/asciiart/animals/fish.htm`

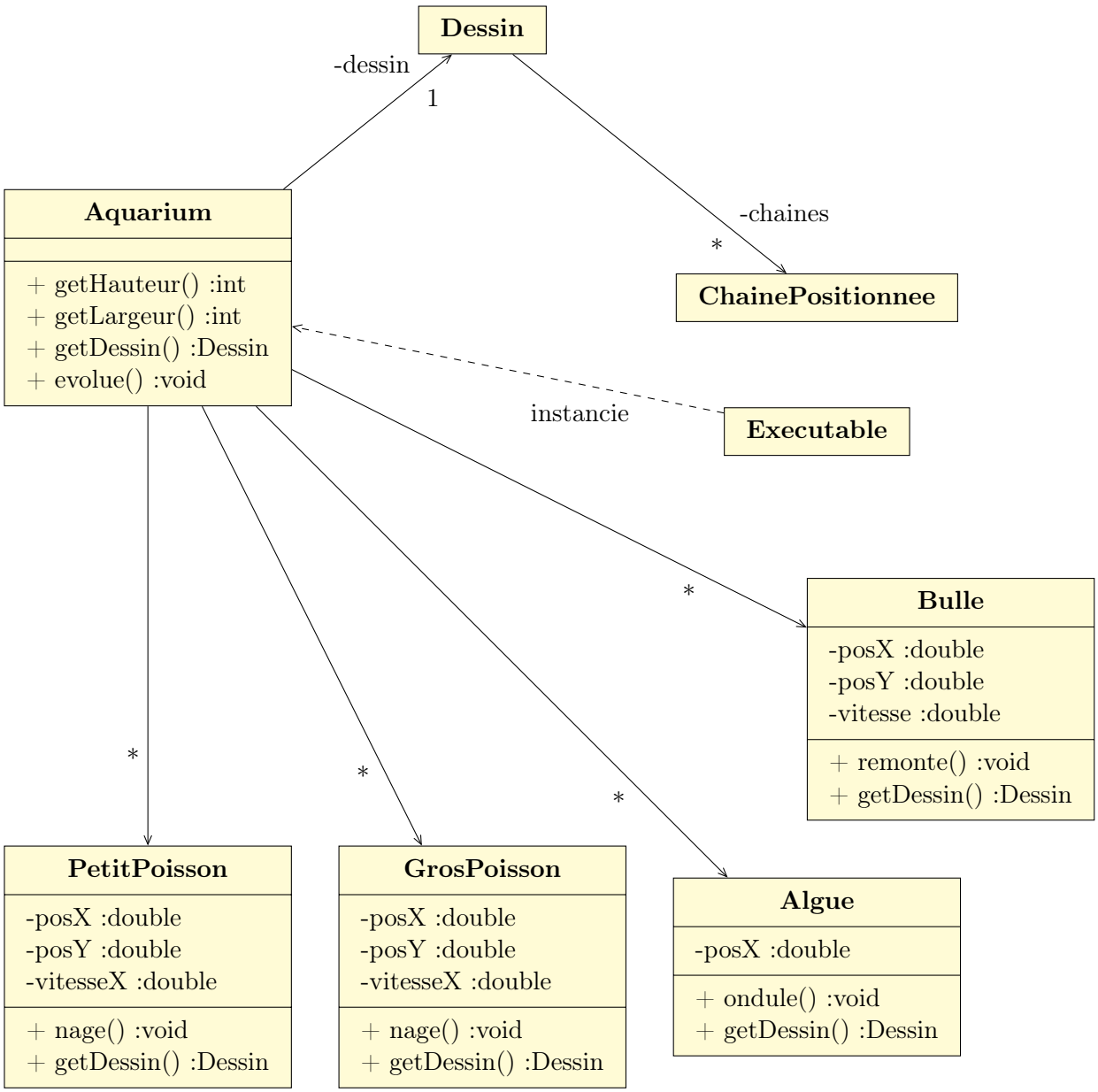
Une première analyse (très perfectible !) a produit le diagramme de classes qui se trouve sur la page suivante. Tu trouveras les classes `ChainePositionnee`, `Dessin` ainsi qu'un `Executable` (dont nul n'est besoin de lire le code) sur Celene.

Quelques précisions :

- Dans la classe `Aquarium` :
  - La méthode `getHauteur()` renvoie la hauteur de l'aquarium
  - La méthode `getLargeur()` renvoie la largeur de l'aquarium
  - La méthode `getDessin()` renvoie le Dessin à afficher
  - La méthode `evolue()` est appelée à fréquence constante pour faire évoluer l'aquarium (déplacer les poissons, faire onduler les algues, remonter les bulles, etc.).
- Un `PetitPoisson` est un poisson dont le dessin ressemble à `<><` . Il se déplace horizontalement (de droite à gauche ou de gauche à droite).
- Un `GrosPoisson` est un poisson qui se déplace horizontalement et dont le dessin ressemble (par exemple) à

```
  ;,//;,     ,;/
 o::::::::;;///
>:::::::::;;\\\
  ''\\\\\'" ';\
```

- Les algues ne se déplacent pas : elles ondulent
- Les bulles de déplacent uniquement du bas vers le haut, jusqu'à arriver à la surface



2.1 A l'aide du code fourni sur Celene, complète sur le diagramme de la feuille réponse les classes `Dessin` et `ChainePositionnee` avec les attributs/constructeurs et méthodes **publiques**.

2.2 Complète le diagramme de la feuille réponse en améliorant l'analyse proposée.

2.3 Implémente le projet `Aquarium`

Prénom Nom : ..... Groupe : .....