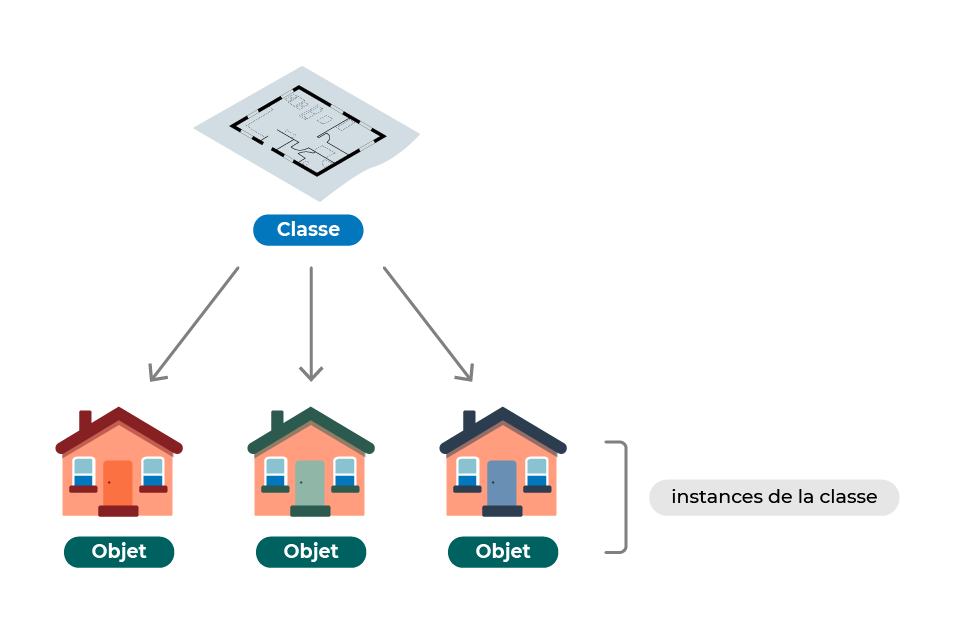
# ***Cours POO***

Représentation d’une classe :



Contenu d’une classe :

* Des **attributs** (c'est le nom que l'on donne aux **variables** contenues dans des classes)
* Des **méthodes** (c'est le nom que l'on donne aux **fonctions** contenues dans des classes).

Droit d’accès :

* **public** : l'attribut ou la méthode peut être appelé depuis l'extérieur de l'objet.
* **private** : l'attribut ou la méthode ne peut pas être appelé depuis l'extérieur de l'objet. Par défaut, tous les éléments d'une classe sont **private**.

Exemple de classe :

Une image contenant texte

Description générée automatiquement

*Source :* [*https://openclassrooms.com/fr/courses/7137751-programmez-en-oriente-objet-avec-c/7673296-tirez-un-maximum-de-ce-cours*](https://openclassrooms.com/fr/courses/7137751-programmez-en-oriente-objet-avec-c/7673296-tirez-un-maximum-de-ce-cours)

# ***2.*** ***Résolution du prosit 1 (bloc POO)***

Ajout d’une ligne ***using namespace std*** pour éviter la répétition de ***std::***   
 std est un objet d’où le ***std::***  
 Utilisation de ***float*** pour le return de la distance.  
  
 Objet est une classe qui possède plusieurs attributs et méthodes.  
 L’objet nous permet d’éviter les récurrences dans le code, de créer des héritages et d’avoir un code plus claire.  
Un objet doit être responsable de ses propres modifications, il est donc interdit de permettre la modification des attribut depuis l’extérieur.  
  
Ficher .h : déclaration des prototype des fonctions ainsi que les attributs.  
  
Diagramme de classe :

* Rectangle correspond à la classe.
* Le nom de la classe doit toujours est en MAJUSCULES.
* **public** et **private** doit être précisé.

Constructeur : méthode permettant d’instancier un objet, il peut avoir des attributs comme paramètres.

Geteur : Permet de récupérer un attribut de la classe.

Seteur : Permet de modifier un attribut de la classe.

Destructeur : Utile si la classe est alloué dynamiquement et permet donc de détruire l’objet.

Utiliser une classe de point.  
L’hypothèse est donc fausse car la distance vaut