Java théorie Q2

***Séance 1 :***

1. Vocabulaire :

Un objet représente un élément du problème à résoudre.

Un type de données OO est déﬁnit par une classe.

Un objet est une instance d’une classe, construit à partir de la définition donnée par la classe, ou appartenant au type défini par la classe. Un objet se caractérise par :

* Son état (ses données) :

Stocké dans des **attributs** (des variables liées à l’objet).

* Son comportement (ce qu’on peut en faire) :

Défini par des **méthodes** (du code).

Membres = attributs + méthodes

/ ! \ pas de static pour les méthodes

1. Constructeur et instanciation :

Instancier un objet c’est le construire en mémoire :

* Lui réserver de l’espace en mémoire (sur le tas)
* Initialiser son état (ses attributs)

L’initialisation est prise en charge par un constructeur.

Pour instancier :

* Utiliser l’opérateur new
* Fournir les paramètres au constructeur

Une classe est un type de référence.

Le constructeur peut vérifier la validité des paramètres.

Null est un littéral de type référence. Il indique qu’il n’y a pas d’objet (référence vers rien). Il ne faut pas confondre null avec «  ».

1. Accès aux membres :

Il est d’accéder aux attributs privés afin d’éviter de donner des valeurs invalides. Il faut alors fournir des méthodes publiques.

1. Accesseurs – mutateurs :

***Définition :*** Accesseur (getter), c’est une méthode donnant la valeur d’un attribut.

Par convention, l’accesseur de attribut est getAttribut, ou isAttribut pour un booléen.

***Définition :*** Mutateur (setter) méthode permettant de modifier l’état d’un objet (modifier un attribut).

Par convention, le mutateur de attribut est setAttribut.

Le mutateur peut vérifier le paramètre. Un objet reste toujours dans un étaat valide.

1. Visibilité :

Privé : n’est visible que de la classe (private)

Paqueté : visible dans toutes les classes du package

Protégé : visibilité liée à l’héritage (protected)

Public : visible dans toutes les classes (public)

1. Constructeur :

Les attributs ont une valeur par défaut :

* 0 pour les nombres
* Null pour les références
* False pour les booléens

Il existe un constructeur par défaut, il n’a pas de paramètre et ne fait rien.

1. Surcharge :

On peut fournir plusieurs constructeurs, ça s’appelle la surcharge (overloading). Ces constructeurs doivent se différencier par nombre ou type des paramètres.

Les méthodes aussi peuvent être surchargées.

1. This :

This est une référence à soi-même.

Elle apparait dans différents contextes :

* Constructeur : this()
* Attribut : this.titre
* Méthode : this.liker()

1. Static :

Trois types de classes :

* Classe utilitaire (Math)
* Classe « objet » (String, Scanner)
* Classe mixte

Un membre static fait référence à une classe et non à une instance. Il est partagé par toutes les instances éventuelles.

L’attribut statique existe en un seul exemplaire. Il est initialisé lors du changement de la classe (une seule fois). L’utilisation courante est faite pour les constantes.

La méthode statique ne peut pas accéder aux membres des instances. L’utilisation courante est celles pour les méthodes non objets.

L’opérateur conditionnel ‘ ? ‘ utilise la valeur booléenne de la première expression pour décider laquelle des deux autres expressions sera évaluée.

On appelle une méthode :

* Statique : par le biais de la classe.
* Non statique : par le biais d’une instance de la classe

L’import static crée un raccourci pour l’accès aux membres statiques.

1. L’encapsulation :

La cohérence de l’objet est assurée par la classe.

* Les attributs sont privés.
* Les méthodes permettant de modifier l’état de l’objet sont publiques ; elles s’assurent que l’objet reste cohérent.

***Séance 2 :*** étude de cas

1. Point :
2. toString :
3. Circle :