

Initiation aux concepts objets en C++

Du C au C++ - Découverte du paradigme objet



Retour en ING1 – Les acquis !

- Déclaration de variables scalaires et complexes
- Conditions et boucles
- Tableaux
- **Allocation dynamique !!**
- **Pointeurs !!**
- Sous programmes
- Structures
- Librairie Allegro 4.X



Sommaire

- I. Du C au C++ : concepts, diagramme de classes et vocabulaire**
- II. Principales différences entre le C et le C++
- III. Classes et objets
- IV. Conteneurs : vecteurs, piles et files
- V. Pointeurs, références et surcharges
- VI. Héritage et polymorphisme



Sommaire (détail)

- I. Du C au C++ : concepts, diagramme de classes et vocabulaire
 - A. Bref historique
 - B. Vocabulaire et concepts objets
 - C. Diagrammes de classe
 - D. Relations entre classes



Objectifs

- Comprendre le concept objet
- Comprendre les relations entre les objets
- Comprendre comment modéliser un problème
- Faire un diagramme de classes



Bref historique du C++

- Auteur : Bjarne Stroustrup
- Première apparition : 1983
- Paradigmes : procédural et orienté objet



Du C au C++

- Structures en C regroupent :
 - Variables de différents types
 - Fonctions (pointeur sur fonction)
- Structures en C++ → Classes
- Objets permettent utilisation des classes



POO – Vocabulaire (1/3)

- POO Késako ?
- Objet = entité de la vie de tous les jours ?
- Attribut = donnée interne d'un objet
- Méthode = fonction ou procédure



P00 – Vocabulaire (2/3)

- Classe = peut être apparenté à un nouveau type
- Instance = « un objet est une instance d'une classe »
- Constructeur = créer un objet
- Destructeur = détruire un objet



PОО – Vocabulaire (3/3)

- Accesseurs
 - « Getter »
 - « Setter »
- **Encapsulation**
 - Protection des données
 - Définition d'une interface standardisée
 - Modularité du code
- **Interface** = fichier d'entête (.h en C)

Diagramme de classes (1/4)

- Exemple :

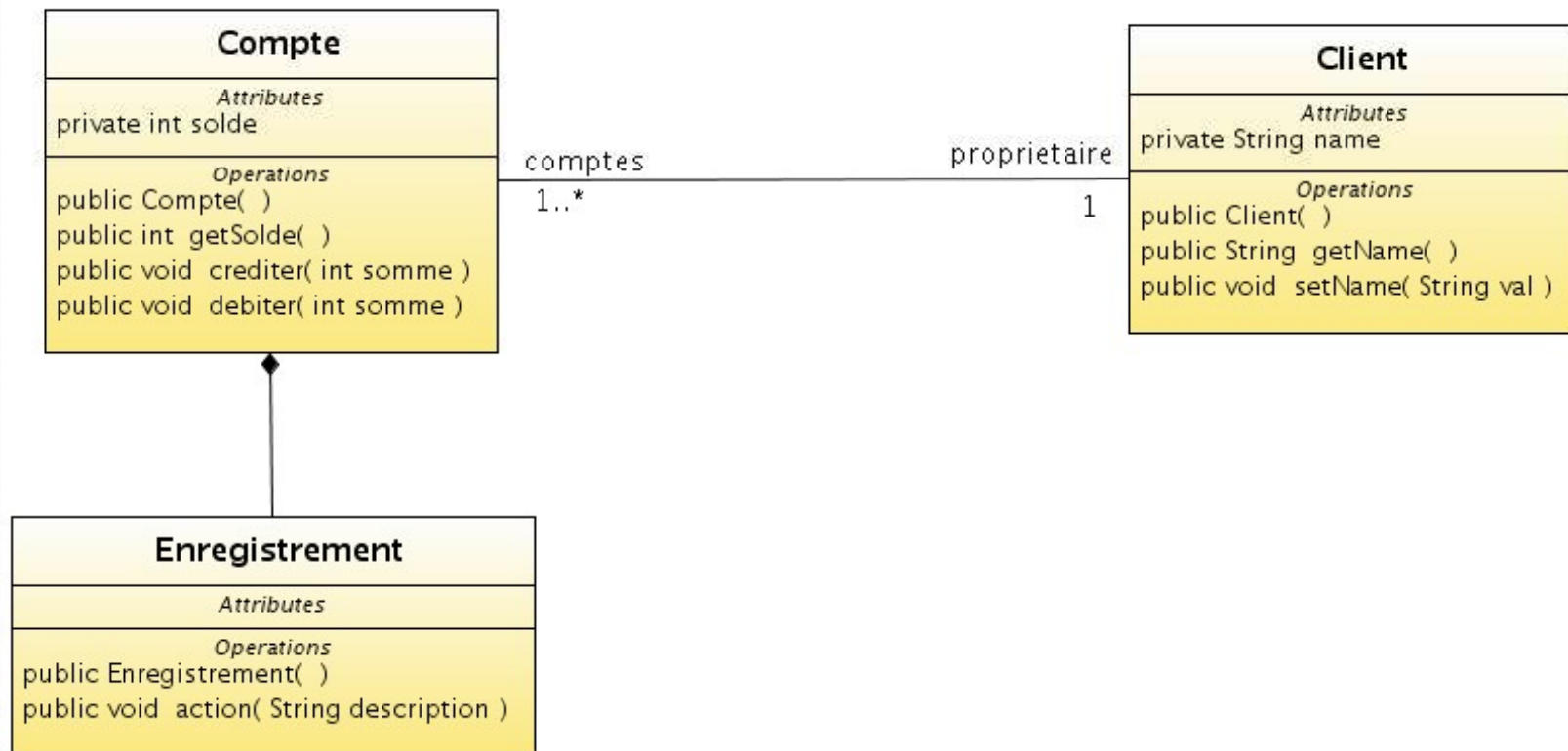


Diagramme de classes (2/4)

- Zoom sur l'entité « Compte »

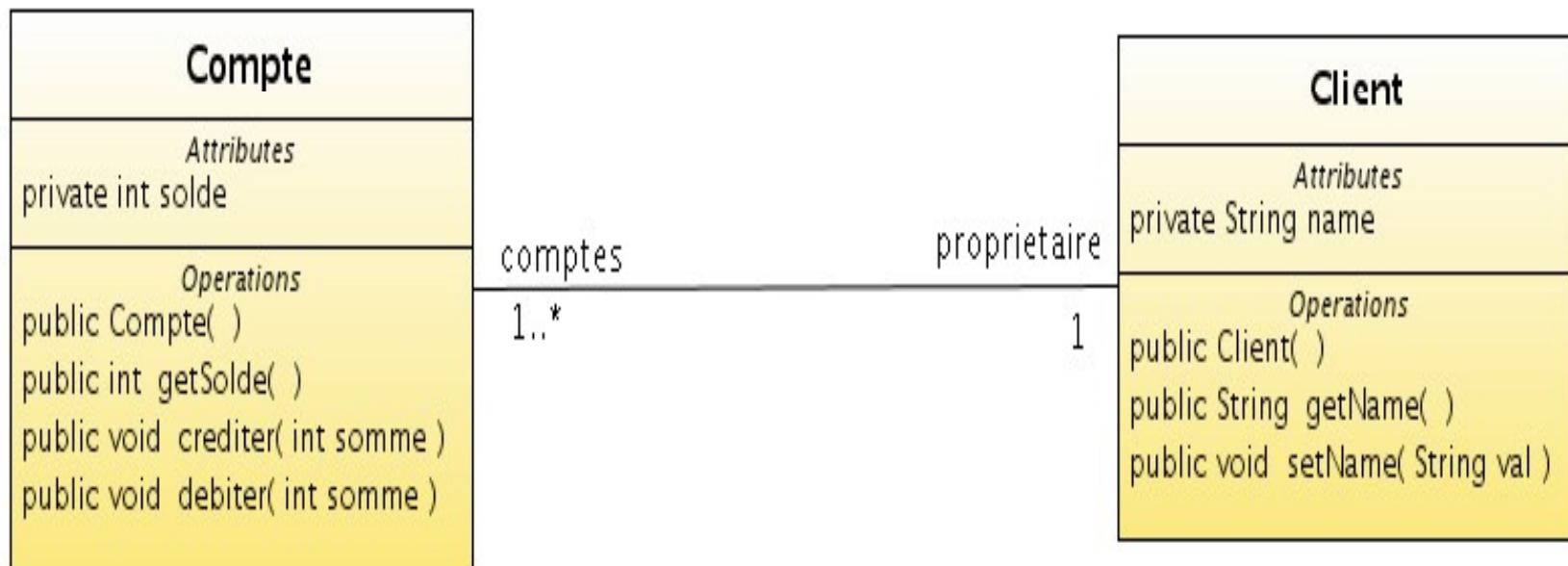
Compte
<i>Attributes</i> private int solde
<i>Operations</i> public Compte() public int getSolde() public void crediter(int somme) public void debiter(int somme)

Diagramme de classes (3/4)

- 3 types de « visibilité » pour les attributs et méthodes
 - « Private » : privée – **Symbole : - (moins)**
 - « Public » : publique – **Symbole : + (plus)**
 - « Protected » : protégée – **Symbole : # (dièse)**
- 3 types de « cardinalité » entre classes
 - Cardinalité 1 – 1
 - Cardinalité 1 – n (ou n – 1)
 - Cardinalité n – n

Diagramme de classes (4/4)

- Exemple de « cardinalité » 1 – n



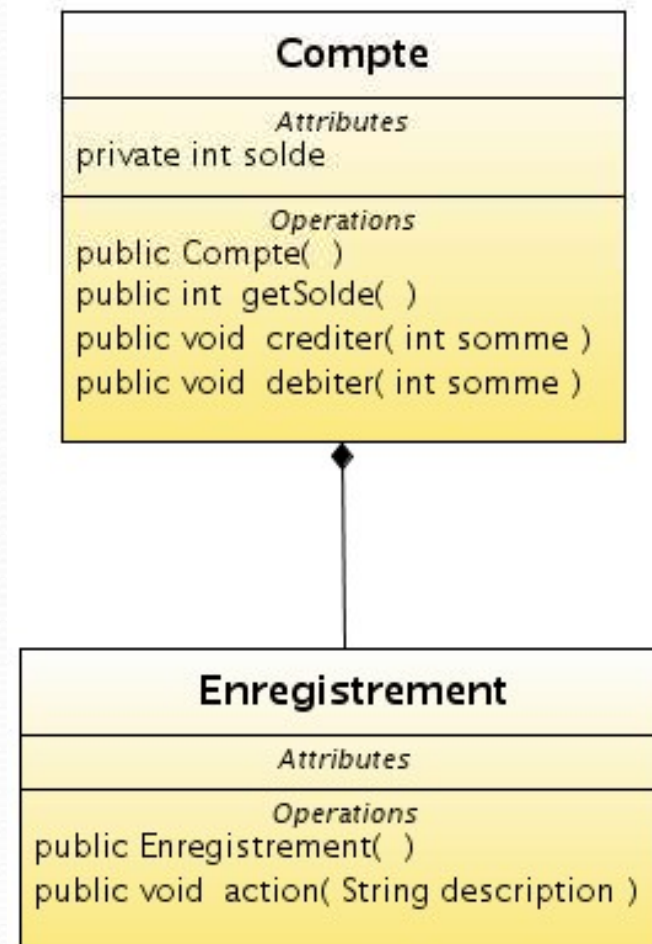


Relations entre classes (1/5)

- 2 types de relation principales entre classes
 - Relation d'agrégation (relation « faible »)
 - Relation de composition (relation « forte »)
- **Attention => Notions subjectives !!**
 - Dépendent du projet et du contexte de l'application

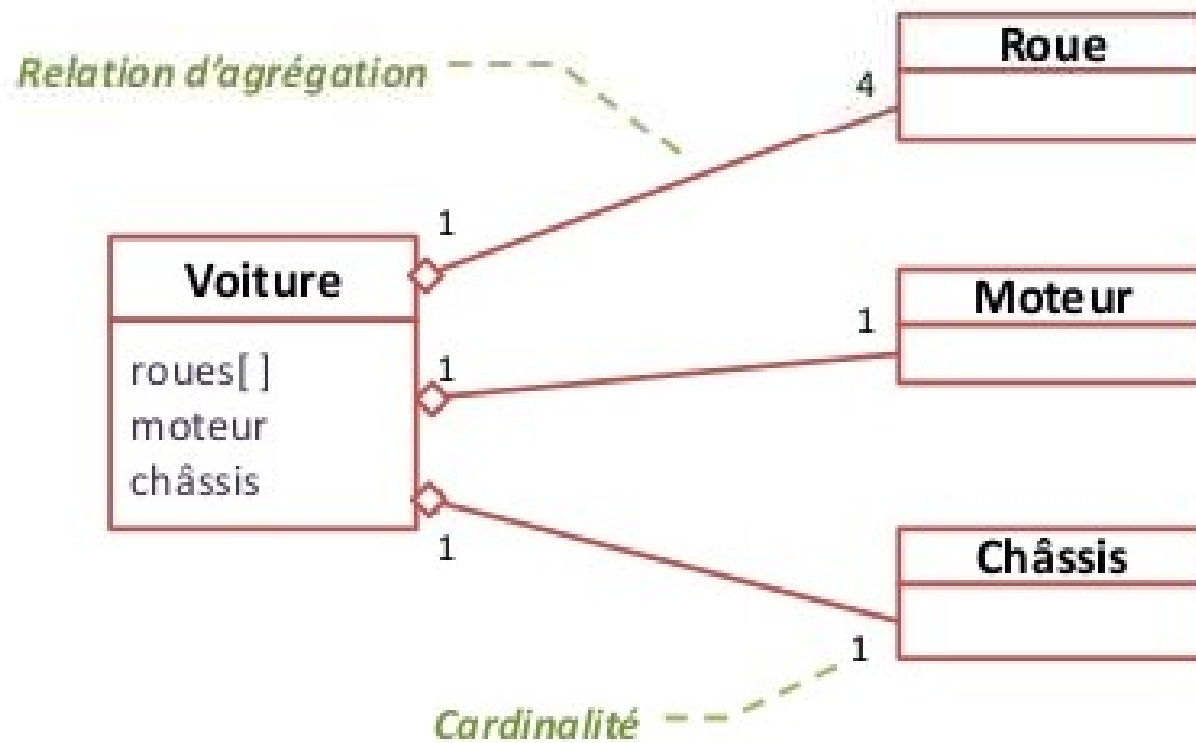
Relations entre classes (2/5)

- Exemple de composition :



Relations entre classes (3/5)

- Exemple de relation d'agrégation



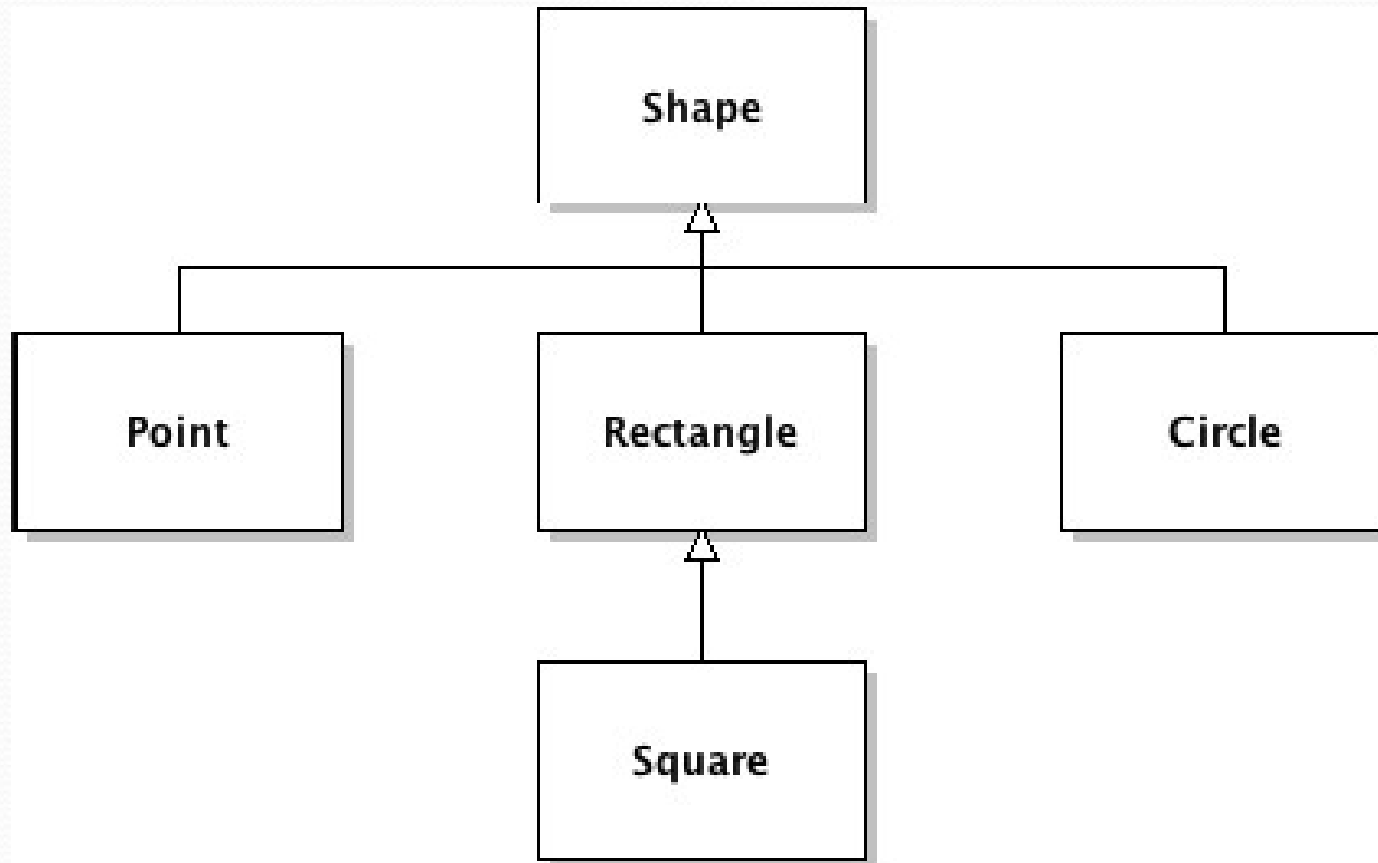


Relations entre classes (4/5)

- Héritage - caractéristiques
 - Relation de parenté entre les classes (mère/fille)
 - Relation transitive
 - Relation non réflexive
 - Relation non symétrique
 - Pas de cycle

Relations entre classes (5/5)

- Exemple d'héritage :





Exercice d'application (1/4)

- Soit l'énoncé suivante :
 - « **Programmer un Mario avec les règles suivantes :**
Mario (le personnage principal) ne peut que marcher, courir et sauter. Il rencontre des ennemies mobiles de différentes tailles, couleurs... en parcourant un niveau 2D avec scrolling horizontal. Il y a au total 8 mondes composés de 4 niveaux chacun. Pour passer d'un niveau à un autre, il faut atteindre la ligne d'arrivée en moins de 360 secondes. »
- Identifier les classes, attributs, méthodes et les relations entre les différentes classes...

Exercice d'application (2/4)

- Classes | méthodes | attributs | Valeur des attributs
 - « Programmer un Mario avec les règles suivantes : Mario (le personnage principal) ne peut que *marcher, courir et sauter*. Il rencontre des ennemies mobiles de différentes tailles, couleurs... en parcourant un niveau 2D avec *scrolling* horizontal. Il y a au total 8 mondes composés de 4 niveaux chacun. Pour passer d'un niveau à un autre, il faut atteindre la ligne d'arrivée en moins de 360 secondes. »
- Mais il peut y avoir des informations cachées !

Exercice d'application (3/4)

- Synthèse :

Classes (nom)	Méthodes (verbe)	Attributs (adj)
Personnage	marcher	nom
	courir	taille
	sauter	
Ennemie	Se déplacer	couleur
		taille
Niveau	charger	numero
	gagner	numero_max
	scroller	temps_max
Monde	changer	numero
		numero_max



Exercice d'application (4/4)

- Relations entre les classes :
 - Un monde est composé d'exactly 4 niveaux
 - Un niveau contient Mario et des ennemies
 - Les ennemies peuvent être de différents types (notion d'héritage...)
 - ...



Prochaine séance

- Principales différences en C et C++