**Modélisation Orientée Objet**

**UML**

**Introduction :**

La conception de projet correspond à l'une des premières phases dans le cycle de vie d'une initiative, au cours de laquelle les idées, processus, ressources et résultats attendus sont planifiés.

La conception d’un projet est une organisation stratégique d’idées, de matériaux et de processus dans le but d’atteindre un objectif. Les chefs de projet s’appuient sur une bonne conception pour éviter les écueils et fournir des paramètres permettant de maintenir les aspects cruciaux du projet, comme le calendrier et le budget.

Le développement des systèmes est une tâche d’une grande envergure et un investissement important pour toute entreprise. La modélisation des systèmes déjà existants ou d’un système à construire est une étape importante du cycle de développement des systèmes. La modélisation permet de visualiser, souvent d’une manière graphique, un système tel qu’il est ou comment nous souhaitons qu’il soit.

La modélisation d’une manière générale, aide à l’élaboration et à la structuration des idées et permet de faciliter la communication.

Un modèle est une abstraction de la réalité. Modéliser consiste à identifier les caractéristiques intéressantes ou pertinentes d’un système dans le but de pouvoir l’étudier du point de vue de ses caractéristiques.

Un bon modèle doit posséder deux caractéristiques essentielles.

* Il doit faciliter la compréhension du système étudié,
* Il doit permettre de simuler le système étudié, il reproduit ses comportements.

1. **Modélisation Orientée Objet :**

La modélisation orientée objet (MOO) est une méthode de conception et de programmation informatique qui repose sur le concept *d’objet*.

Un objet représente une entité du monde réel qui possède des caractéristiques (appelées attributs) et des comportements (appelés méthodes). Cette approche permet de structurer le code de manière à mieux refléter le fonctionnement des systèmes complexes.

Voici quelques concepts fondamentaux de la modélisation orientée objet :

***Classe :*** Une classe est une sorte de plan ou de modèle qui définit la structure et les comportements d'un type d'objet. Par exemple, si nous parlons d'une classe "Voiture", elle pourrait avoir des attributs tels que la marque, le modèle, la couleur, etc., et des méthodes telles que "démarrer", "arrêter", etc.

***Objet :*** Un objet est une instance d'une classe. Par exemple, une voiture spécifique comme "MaVoiture" serait une instance de la classe "Voiture".

***Attributs :*** Les attributs représentent les caractéristiques d'un objet. Ils définissent l'état de l'objet. Dans l'exemple de la classe "Voiture", les attributs pourraient être "marque", "modèle", "couleur", etc.

***Méthodes :*** Les méthodes sont des fonctions ou des procédures associées à une classe. Elles définissent les actions que l'objet peut effectuer. Par exemple, dans la classe "Voiture", les méthodes pourraient être "démarrer", "arrêter", "changer de vitesse", etc.

1. **Le langage « UML » :**

Le **Langage de Modélisation Unifié**, de l'anglais ***U****nified****M****odeling****L****anguage* (**UML**), est un [langage](https://fr.wikipedia.org/wiki/Langage) de modélisation graphique à base de [pictogrammes](https://fr.wikipedia.org/wiki/Pictogramme) conçu comme une méthode normalisée de visualisation dans les domaines du [développement logiciel](https://fr.wikipedia.org/wiki/D%C3%A9veloppement_logiciel) et en [conception orientée objet](https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_objet).

L'UML est une synthèse de langages de modélisation objet antérieurs : [Booch](https://fr.wikipedia.org/wiki/M%C3%A9thode_Booch), [OMT](https://fr.wikipedia.org/wiki/Object_Modeling_Technique), [OOSE](https://fr.wikipedia.org/wiki/OOSE). Principalement issu des travaux de [Grady Booch](https://fr.wikipedia.org/wiki/Grady_Booch), [James Rumbaugh](https://fr.wikipedia.org/wiki/James_Rumbaugh) et [Ivar Jacobson](https://fr.wikipedia.org/wiki/Ivar_Jacobson), UML est à présent un standard adopté par l'[Object Management Group](https://fr.wikipedia.org/wiki/Object_Management_Group) (OMG). UML 1.0 a été normalisé en janvier 1997; UML 2.0 a été adopté par l'OMG en juillet 2005. La dernière version de la spécification validée par l'OMG est UML 2.5.1 (2017).

UML est un langage de modélisation. La version actuelle, UML 2.5, propose 14 types de diagrammes. Les diagrammes sont représentés sous deux types de vue : d'un point de vue statique ou structurelle du domaine avec les diagrammes de structure (Structure Diagrams). D'un point de vue dynamique avec les diagrammes de comportement (Behavior Diagrams) et les diagrammes d’interactions (Interaction Diagrams).

**Définition d’un diagramme**

Un diagramme UML est une représentation graphique, qui s'intéresse à un aspect précis du modèle. Chaque type de diagramme UML possède une structure (les types des éléments de modélisation qui le composent sont prédéfinis).

Un type de diagramme UML offre toujours la même vue d'un système (il véhicule une sémantique précise). Combinés, les différents types de diagrammes UML offrent une vue complète des aspects statiques et dynamiques d'un système.

**Les *diagrammes de structure* (*structure diagrams*) ou *diagrammes statiques* (*static diagrams*) rassemblent :**

* [***Diagramme de classes***](https://fr.wikipedia.org/wiki/Diagramme_de_classes)**(*class diagram*) :** représentation des [classes](https://fr.wikipedia.org/wiki/Classe_(informatique)) intervenant dans le système.
* [***Diagramme d'objets***](https://fr.wikipedia.org/wiki/Diagramme_d%27objets)**(*object diagram*) :** représentation des instances de classes (objets) utilisées dans le système.
* [***Diagramme de composants***](https://fr.wikipedia.org/wiki/Diagramme_de_composants)**(*component diagram*) :** représentation des composants du système d'un point de vue physique, tels qu'ils sont mis en œuvre ([fichiers](https://fr.wikipedia.org/wiki/Fichier_(informatique)), [bibliothèques](https://fr.wikipedia.org/wiki/Biblioth%C3%A8que_logicielle), [bases de données](https://fr.wikipedia.org/wiki/Base_de_donn%C3%A9es)…)
* [***Diagramme de déploiement***](https://fr.wikipedia.org/wiki/Diagramme_de_d%C3%A9ploiement)**(*deployment diagram*) :** représentation des éléments matériels ([ordinateurs](https://fr.wikipedia.org/wiki/Ordinateur), [périphériques](https://fr.wikipedia.org/wiki/P%C3%A9riph%C3%A9rique_(informatique)), [réseaux](https://fr.wikipedia.org/wiki/R%C3%A9seau_informatique), systèmes de stockage…) et la manière dont les composants du système sont répartis sur ces éléments matériels et interagissent entre eux.
* [***Diagramme des paquets***](https://fr.wikipedia.org/wiki/Diagramme_des_paquetages)**(*package diagram*) :** représentation des dépendances entre les paquets (un paquet étant un conteneur logique permettant de regrouper et d'organiser les éléments dans le modèle UML), c'est-à-dire entre les ensembles de définitions.
* [***Diagramme de structure composite***](https://fr.wikipedia.org/wiki/Diagramme_de_structure_composite)**(*composite structure diagram*) :** représentation sous forme de [boîte blanche](https://fr.wikipedia.org/wiki/Bo%C3%AEte_blanche) des relations entre composants d'une classe (depuis UML 2.x).
* [***Diagramme de profils***](https://fr.wikipedia.org/wiki/Diagramme_de_profils)**(*profile diagram*) :** spécialisation et personnalisation pour un domaine particulier d'un meta-modèle de référence d'UML (depuis UML 2.2).

**Diagrammes de comportement :**

**Les diagrammes de comportement (behavior diagrams) rassemblent :**

* [**Diagramme des cas d'utilisation**](https://fr.wikipedia.org/wiki/Diagramme_des_cas_d%27utilisation)**(use-case diagram)** : représentation des possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système), c'est-à-dire de toutes les fonctionnalités que doit fournir le système.
* [**Diagramme états-transitions**](https://fr.wikipedia.org/wiki/Diagramme_%C3%A9tats-transitions)**(state machine diagram) :** représentation sous forme de machine à états finis du comportement du système ou de ses composants.
* [**Diagramme d'activité**](https://fr.wikipedia.org/wiki/Diagramme_d%27activit%C3%A9)**(activity diagram) :** représentation sous forme de flux ou d'enchaînement d'activités du comportement du système ou de ses composants.

**Diagrammes d'interaction ou diagrammes dynamiques :**

Les diagrammes d'interaction (interaction diagrams) ou diagrammes dynamiques (dynamic diagrams) rassemblent :

* [**Diagramme de séquence**](https://fr.wikipedia.org/wiki/Diagramme_de_s%C3%A9quence)**(sequence diagram) :** représentation de façon séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs.
* [**Diagramme de communication**](https://fr.wikipedia.org/wiki/Diagramme_de_communication)**(communication diagram) :** représentation de façon simplifiée d'un diagramme de séquence se concentrant sur les échanges de messages entre les objets (depuis UML 2.x).
* [**Diagramme global d'interaction**](https://fr.wikipedia.org/wiki/Diagramme_global_d%27interaction)**(interaction overview diagram) :** représentation des enchaînements possibles entre les scénarios préalablement identifiés sous forme de diagrammes de séquences (variante du [diagramme d'activité](https://fr.wikipedia.org/wiki/Diagramme_d%27activit%C3%A9)) (depuis UML 2.x).
* [**Diagramme de temps**](https://fr.wikipedia.org/wiki/Diagramme_de_temps)**(timing diagram) :** représentation des variations d'une donnée au cours du temps (depuis UML 2.3).

