# Annexe 1 - Qu'est-ce que la programmation orientée objet ?

1. C'est facile de savoir ce qu'est un objet : Crayon, tableau, un ecran

…ou ce qui n'est pas un objet : émotions, 37 % du crayon, air

On voit le monde qui nous entoure en termes d'objets et de leurs relations entre eux, notre mémoire est organisée de cette manière. On va reproduire la même chose avec la programmation orientée objet : représenter la réalité le plus fidèlement possible dans nos programmes

1. Fonctionnement d'une voiture 🡪 simuler la conduite d'une auto dans un jeu vidéo



*Tiré de Nascar Heat 4, jeu de Playstation*

Différents objets :Volant, rétroviseur, voiture

* fabriqués à partir d'un plan : une classe (programme)
* communiquant entre eux : s’envoie des messages (méthode)
* dans un but commun : jeu/voiture fonctionne

1. Qu’est-ce qu’un objet dans le cadre d'un projet de programmation OO ?

OBJET : capsule d’informations

Attributs qui représentent l’état de l’objet (variables d’instance) Toutes les variables qui distingue une voiture d’une autre voiture. Ex : la vitesse, le nombre de tours, etc.

Comportements : ce qu’il peut faire de lui-même ou en réaction a un message externe. La voiture roule et si une autre voiture la frappe elle va alors réagir. (Méthode)

Exemples :

|  |  |  |
| --- | --- | --- |
| Objets | Données / attributs | Comportements |
|  |  |  |
| Jauge à essence | Int qteEssence; | * Public int transmettreTableauDEBord()   Return qteEssence |
| Conducteur | Int experience  Int age | * Tourner à gauche * Freine |
| Moteur | Int Rpm  String marque  Piston [] tab  Soupap[] tab | Envoie un message a la transmission(puissance generee)  Envoie un message au tableau de bord contenant le nb de Rpm |

1. Regardez le vidéo ( bowling.mp4 sur LEA ) et déterminez les différents objets, leurs attributs et leurs comportements :

etat

La taille

Sa position x, y

Vitesse

comportement

avancer

Fonction qui fait le strike secret

1. Un objet seul est très rarement utile; plusieurs objets interagissent entre eux dans un but commun :

changerVitesse(4)

Objet TransmissionManuelle Objet Conducteur

variable int vitesse

méthode void changerVitesse( int nlleVitesse )

* Lorsque l'objet Conducteur veut que l'objet TransmissionManuelle change de vitesse, l'objet Conducteur envoie un message à l'objet TransmissionManuelle
* Cet envoi de message entraîne l'objet TransmissionManuelle à réaliser une opération sur lui-même; il regarde le message reçu et choisit la méthode à appliquer ( ici c'est la méthode changerVitesse )
* Il a besoin d'un paramètre, l'objet TransmissionManuelle ne peut pas déterminer par lui-même quelle vitesse choisir, c'est pourquoi le message comprend la décision extérieure à l'objet.
* La méthode changerVitesse d'un objet TransmissionAutomatique n'aurait donc pas besoin de paramètre, pourquoi ? La methode ce change d elle-même sans intervention exterieure

1. Chronologie des langages orientés objet

*Simula 67* : langage de simulation, créé pour résoudre des problèmes mathématiques trop complexes pour des équations.

*C++* : des ajouts à C notamment pour faire des applications orientés objet.

*Eiffel, Ada (* 1er langage de programmation à l'UQAM durant longtemps, très peu utilisé en entreprise, formateur mais complexe )

*SmallTalk* : pur OO ( tout est un objet, même les variables contenant des nombres ) tandis qu'en Java on a quand même des types prédéfinis : int, double, char …

('95) *Delphi* : langage objet inspiré de la syntaxe du langage Pascal, s'adapte à la plateforme .NET

('95) *Java* :

* Approche OO de SmallTalk
* Syntaxe C++ simplifiée ( pas de pointeurs )
* À l'origine devait s'appeler Oak ( déjà pris )
* Créé par James Gosling, canadien d'origine : <http://nighthacks.com/roller/jag/>
* Java a profité de l'engouement pour internet ( multiplateforme )

La très grand majorité des langages actuels ont une vision orientée objet : Python ( 1991 env. ), C# ( 2000 ), PHP, Kotlin, Objective-C ....

Le développement web et mobile est également conçu à l'aide d'objets et de classes.