

Notion de programmation orientée objet

Christophe Viroulaud

Terminale - NSI

Lang 01



Contexte
historique

Programmation impérative

Programmation orientée
objet

Classe

Définition

Modélisation

Instanciation

Découvrir un nouveau paradigme de programmation.

Contexte historique

Programmation impérative

Programmation orientée
objet

Classe

Définition

Modélisation

Instanciation

1. Contexte historique

1.1 Programmation impérative

1.2 Programmation orientée objet

2. Classe

Contexte historique - Programmation impérative

Notion de
programmation
orientée objet

Contexte
historique

Programmation impérative

Programmation orientée
objet

Classe

Définition

Modélisation

Instanciation

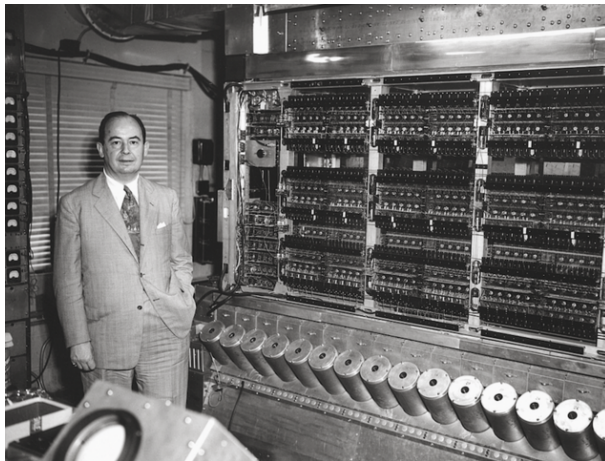


Figure 1 – **fin** année 40 : John von Neumann devant l'EDVAC

À retenir

La programmation impérative (ou procédurale) peut être résumée ainsi :

programme = algorithme + structure de données

Afin de traiter les données, on appréhende le problème en raisonnant de façon logique, d'un état initial vers un état final.

source : chapitre construit d'après un travail de Sylvain Cherrier

Contexte
historique

Programmation impérative

Programmation orientée
objet

Classe

Définition

Modélisation

Instanciation

Observation

Le déroulement *logique* d'un programme procédural :

- ▶ les données *entrent* dans l'algorithme,
- ▶ des actions définies sont appliquées,
- ▶ le programme termine.

Avantages

- ▶ proche du fonctionnement de la machine,
- ▶ déroulement linéaire,
- ▶ raisonnement logique.

Inconvénients

- ▶ forte dépendance algorithme $\leftrightarrow$ données,
- ▶ peu réutilisable,
- ▶ peu d'interfaçage entre applications.

Contexte
historique

Programmation impérative

**Programmation orientée
objet**

Classe

Définition

Modélisation

Instanciation

1. Contexte historique

1.1 Programmation impérative

1.2 Programmation orientée objet

2. Classe

À retenir

La POO veut réunir les données et leur traitement en une unité cohérente et maintenable.

Exemple

Sur un site d'achat, un utilisateur peut :

- ▶ se connecter,
- ▶ acheter,
- ▶ mettre un commentaire,
- ▶ ...

Contexte
historique

Programmation impérative

Programmation orientée
objet

Classe

Définition

Modélisation

Instanciation

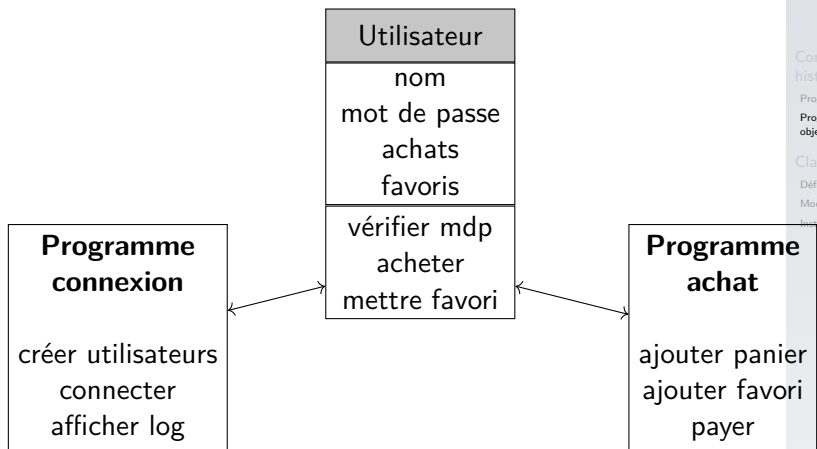


Figure 2 – L'objet **Utilisateur** contient les données le caractérisant et leurs traitements possibles.



Figure 3 – **1962** : Premier langage qui implémente le modèle de classes de Hoare, prémisse de la programmation objet

Contexte
historique

Programmation impérative

Programmation orientée
objet

Classe

Définition

Modélisation

Instanciation

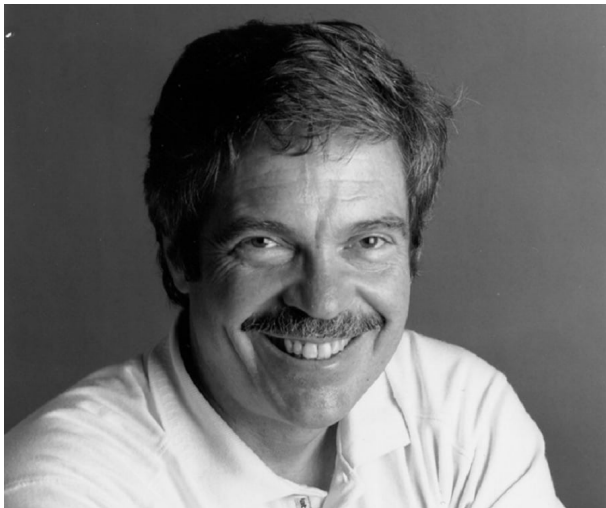


Figure 4 – **années 70** : Alan Kay est considéré comme le père de la POO (Flex puis Smalltalk).

Contexte historique

Programmation impérative

Programmation orientée
objet

Classe

Définition

Modélisation

Instanciation

1. Contexte historique

2. Classe

2.1 Définition

2.2 Modélisation

2.3 Instanciation

À retenir

Une **classe** est une modélisation d'un élément du monde réel. Une classe possède :

- ▶ **des attributs** : il s'agit des caractéristiques spécifiques de l'objet (données),
- ▶ **des méthodes** : il s'agit des fonctionnalités que peut réaliser l'objet (traitements et services).

Contexte
historique

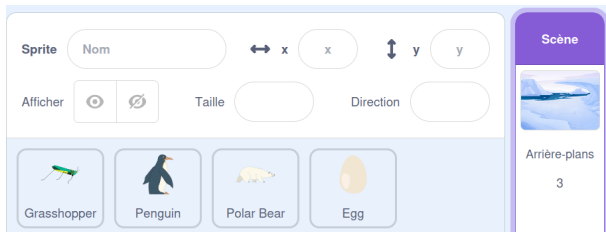
Programmation impérative
Programmation orientée
objet

Classe

Définition

Modélisation

Instanciation



Remarque

On utilise souvent indifféremment les termes **classe** et **objet**.

1. Contexte historique

2. Classe

2.1 Définition

2.2 Modélisation

2.3 Instanciation

Contexte
historique

Programmation impérative

Programmation orientée
objet

Classe

Définition

Modélisation

Instanciation

Utilisateur

► Attributs :

- nom,
- mail,
- mot de passe,
- liste de commandes,
- panier,
- favoris

► Méthodes :

- vérifier le mot de passe,
- regarder les anciennes commandes,
- ajouter un favori,
- ajouter dans le panier

Une modélisation possible

Contexte
historique

Programmation impérative

Programmation orientée
objet

Classe

Définition

Modélisation

Instanciation

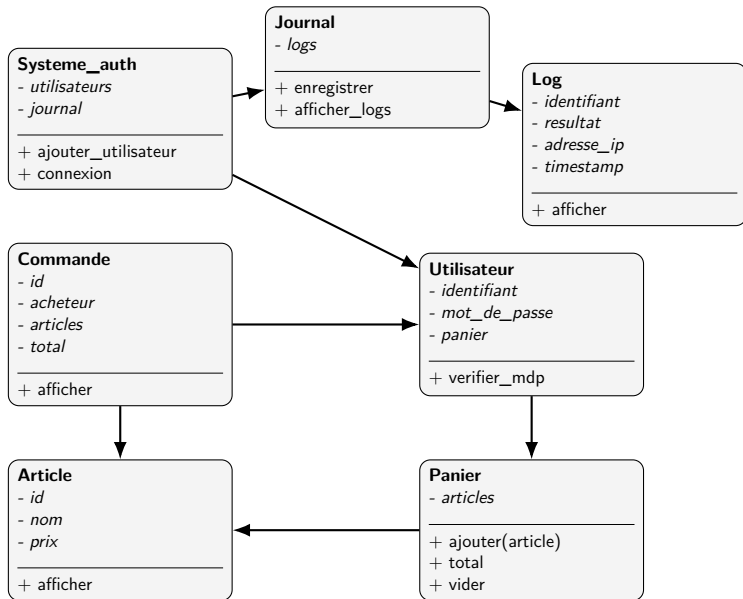


Figure 5 – La modélisation de la problématique consiste à définir les objets et leurs interactions.

1. Contexte historique

2. Classe

2.1 Définition

2.2 Modélisation

2.3 Instanciation

Contexte historique

Programmation impérative

Programmation orientée
objet

Classe

Définition

Modélisation

Instanciation

À retenir

- ▶ Un **objet (ou classe)** est une structure modèle abstraite.
- ▶ Une **instance** est une construction *concrète*, produite à partir de l'objet modèle. L'instance est manipulable dans le programme.

Contexte
historique

Programmation impérative

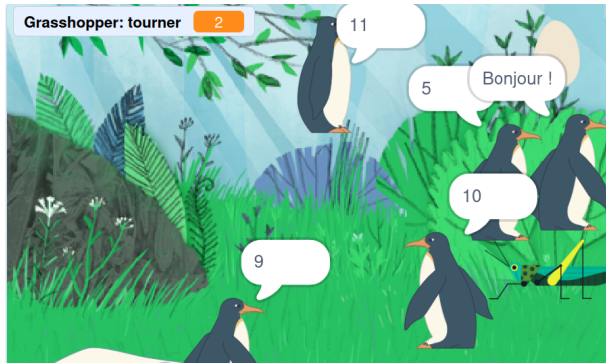
Programmation orientée
objet

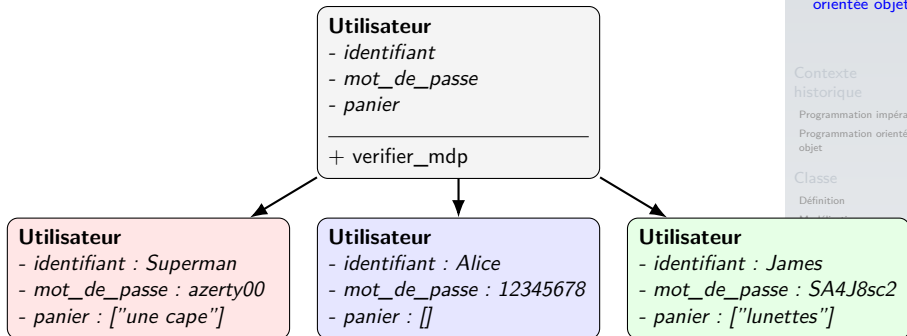
Classe

Définition

Modélisation

Instanciation





Observation

- Chaque utilisateur réel est **une instance** de la classe **Utilisateur**.
- Chaque instance possède des valeurs propres pour ses attributs.