



Ministère de l'enseignement supérieur et de la recherche scientifique  
Université Djillali BOUNAAMA - Khemis Miliana (UDBKM)  
Faculté des Sciences et de la Technologie  
Département de Mathématiques et d'Informatique



## Chapitre 1

# INTRODUCTION

**MI-GLSD-M1-UEM213 : Paradigmes de langages de Programmation**

**Nouredine AZZOUZA**

n.azzouza@univ-dbkm.dz

# Plan du Cours

**1. Objectifs**

**2. Logiciel**

**3. Langage de Programmation**

**4. Paradigme de Programmation**

**4. Concepts modernes de programmation**

The background of the slide is split vertically. The left side is white, and the right side is dark grey. On the right side, there is a faint, high-angle photograph of a hand holding a black pen, writing checkmarks on a list of items. The checkmarks are drawn in a simple, bold style. The text 'Objectifs du Cours' is overlaid on this background. The word 'Objectifs' is in blue, and 'du Cours' is in white.

# = Objectifs du Cours

# Objectifs

- ✓ Se familiariser avec diverse paradigmes de programmation
- ✓ Connaître les principes fondamentaux de diverse paradigmes
- ✓ Etudier les différences principales des paradigmes de programmation
- ✓ Pouvoir développer des programmes en
  - ✓ en utilisant **diverses paradigmes**.
  - ✓ en utilisant **de bonnes pratiques** de programmation



# Logiciel

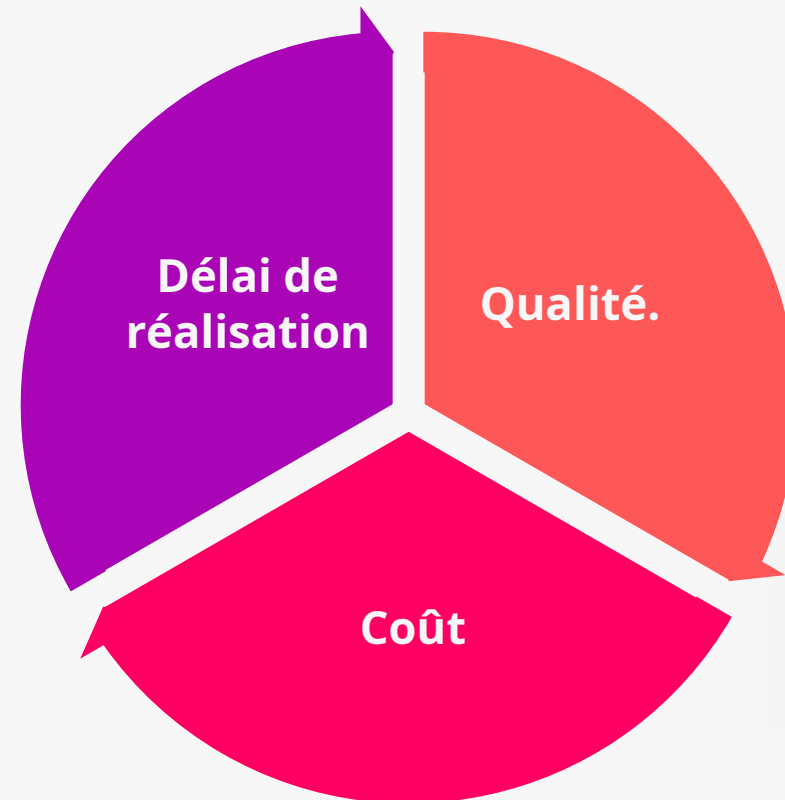


## Définition

- ✓ C'est un ensemble de programmes et de règles relatives à un traitement de l'information.

## Caractéristiques

- ✓ un logiciel peut être considéré comme un produit manufacturé, c'est à dire astreint aux exigences de tout produit.



# Qualités

- ✓ Facilité d'expression et d'utilisation.
- ✓ Facilité de lisibilité et de compréhension.
- ✓ Maintenance aisée et moins coûteuse « importance économique ».
- ✓ Réutilisation sous toutes ses formes (transportabilité).
- ✓ Efficacité des codes obtenus : capacité des codes, rapidité d'exécution, compilation séparée.
- ✓ Utilisation de peu de ressources, c'est-à-dire peu de place mémoire et du temps de calcul.
- ✓ Fiabilité et robustesse des programmes « protection, gestion des anomalies ».

# Langages de

# Programmation





## Définition

- ✓ c'est une **abstraction** des opérations réalisable sur ordinateur.
- ✓ un langage est un ensemble de caractères, de symboles et de **règles** qui permettent de les assembler en vue de communiquer avec un partenaire (Ordinateur). [LAROUSSE]

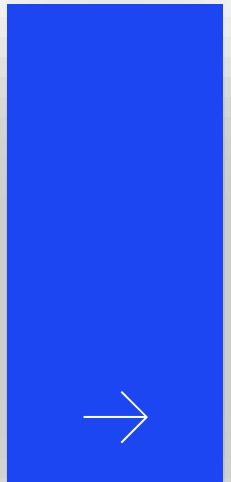
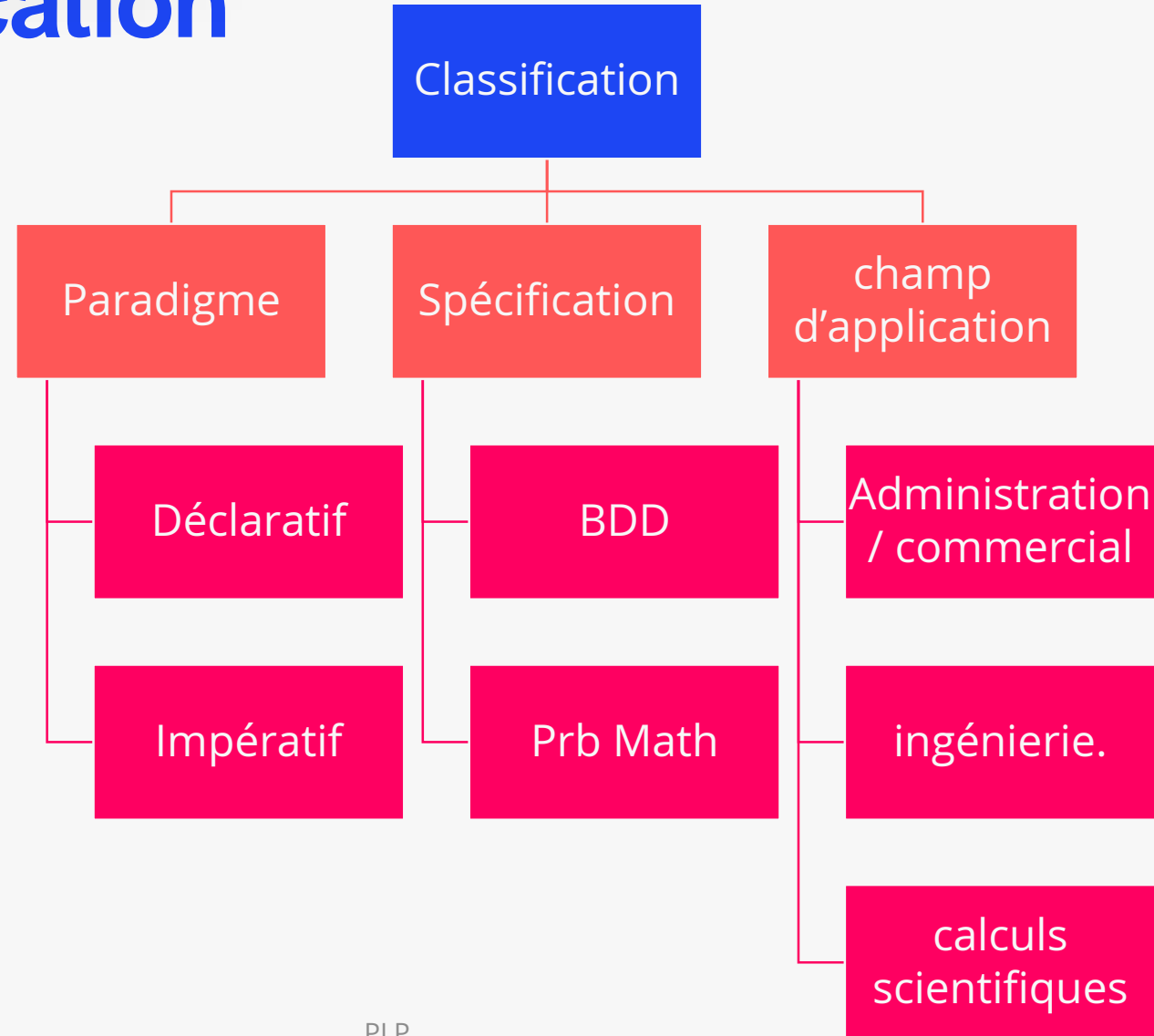


# Application

- ✓ elle est orientée vers un grand public des développeurs leur servant comme outil pour:
  - ❑ s'exprimer facilement.
  - ❑ structurer ses idées.
  - ❑ apprendre de nouveaux langages de programmation.



# Classification



## Description

### i. Syntaxe Formelle :

- ❑ elle concerne la « forme » et la structure des expressions pouvant être construites à l'aide de langage.
- ❑ La syntaxe d'un langage de programmation définit l'ensemble des règles qui sont admissibles et elle est définie par une grammaire

### ii. Sémantique formelle:

- ❑ concerne l'aspect sens et signification à attribuer aux constructions linguistiques qu'offre le langage.
- ❑ Cet aspect souvent exprimé de manière informelle via le langage pose un problème d'ambiguïté, d'inconsistance et d'incomplétude

# Sémantique formelle

## opérationnelle

- exprime le comportement exact de la machine lors de l'exécution d'un programme, elle est orientée développeur de compilateur

## axiomatique

- un ensemble d'axiomes qui expriment les propriétés du langage. Elle est orientée programmeurs

## notationnelle

- c'est une traduction des expressions du langage en un ensemble de dénотations. Elle est orientée concepteurs de langages de programmation



# Approches de programmation

## Approche impérative

- basée sur le concept d'affectation et du séquençement
- L'utilisation des structures de contrôle
- Exemple : Pascal, C

## Approche déclarative

- absence de séquençement explicite et d'affectation
- le programme décrit ce qui doit être calculé
- Exemple: LISP, PROLOG





## Définition

- ✓ est un modèle théorique dépensé qui oriente la recherche et la réflexion scientifique (Larousse 1997).
- ✓ un paradigme est aussi une façon d'aborder un problème de programmation à l'aide d'un type de langage qui supporte bien certains mécanismes d'abstraction.





# Classification

## 1. impératifs

- le programme est constitué d'une suite d'énoncés exécutés séquentiellement.
- **Exemple** : ADA, PASCAL, FORTRAN, COBOL, MODULA, C

## 2. fonctionnels

- le programme est vu comme une transformation de données à fin d'obtenir la sortie désirée.
- **Exemple** : LISP, ML, HOPE, MIRANDA, HASKELL

## 3. logiques

- le programme décrit les propriétés de la solution recherchée , et un mécanisme d'inférence propose des solutions.
- **Exemple** : PROLOG, GODEL

## 4. orienté objet

- le programme se décompose en un ensemble d'objets interagissant entre eux par l'échange de messages.
- **Exemple** : SMALTALK, C++, JAVA

## 5. Concurrent

- le programme autorise la répartition des tâches à accomplir en un ensemble de plusieurs processus parallèles.
- **Exemple** : ADA 95



# Concepts modernes

# de Programmation

STARTUP & DEVELOPMENT



## Modularité

- ✓ un mécanisme puissant de structuration des programmes.
- ✓ C'est le concept central du génie logiciel (GL) motivé par :
  - ❑ Vaincre la complexité.
  - ❑ Augmenter la réutilisation.



# Abstraction

- ✓ se concentrer sur les qualités essentielles des choses plutôt que leurs réalisations concrètes détaillées
- ✓ ceci va nous permettre de maîtriser la complexité.



# Traitement des exceptions

- ✓ mécanisme de détection des anomalies et de récupération tout en interdisant leur propagation.



# Concurrence et parallélisme

- ✓ **Programme séquentiel** : c'est un programme qui est défini par une séquence d'actions c'est-à-dire une instruction après l'autre
- ✓ **Programme concurrent** : un programme qui contient deux ou plusieurs processus qui communiquent entre eux par l'intermédiaire de variables partagées ou par l'échange de messages et des signaux.

