

# Paradigmes de programmation — Introduction

Xavier Crégut  
<Prénom.Nom@enseeiht.fr>

ENSEEIH  
Sciences du Numérique

## Avant de commencer...

**Question :** Que vous évoquent les termes :

- 1 programmation impérative
- 2 programmation fonctionnelle
- 3 programmation déclarative (en particulier programmation logique)
- 4 programmation objet
- 5 programmation réactive

## Information sur le Web

- **Site de l'UE sur Toulouse :**

`http://cregut.perso.enseeiht.fr/ENS/2018-cnam-utc503/index.html`

Ou plus court : `http://cregut.perso.enseeiht.fr/` puis UTC503.

- supports de cours
- sujet d'exercices (TD et TP) et propositions de solutions

- **Liste de diffusion :** `utc503@listes-diff.enseeiht.fr`

Inscription sur : `https://listes-diff.enseeiht.fr/sympa/info/utc503`

- pour recevoir des informations sur le cours
- pour poser des questions
- pour échanger entre auditeurs
- pour recevoir les résultats de l'UE

- **Site de Paris :** `http://formation.cnam.fr/rechercher-par-discipline/paradigmes-de-programmation-1004473.kjsp`

## Objectifs du module

**Objectifs** : Connaître et approfondir les principaux paradigmes de programmation : impératif, logique, fonctionnel, réactif, objet ; savoir les mettre en œuvre ; comprendre leurs différences.

**Compétences visées** : Pouvoir aborder un nouveau langage de programmation ou une nouvelle bibliothèque en reconnaissant les usages dans ceux-ci des principaux paradigmes. La plupart des langages de programmation actuels étant hybrides, et s'ouvrant de plus en plus au paradigme fonctionnel, les connaissances dans un paradigme seront utilisables au-delà de celui-ci.

**Pré-requis** : Connaître un langage de programmation avancé, comme java, et maîtriser les notions d'algorithme, de procédure et fonction, d'objet, de méthode, et d'héritage.

# Organisation des séances

**Où :** ENSEEIHT, B205

- Des badges d'accès seront fournis via l'IPST (sinon passage par l'accueil)
- Sortie libre par les tourniquets (ne pas prendre celui qui mène au parking, piège !)

**Quand :** lundi de 18 à 21h

- du 5 octobre 2018 au 3 décembre 2018
- pas de cours le 29 octobre 2018

**Séance type :**

- 1 Retour sur les exercices précédents. Questions.
- 2 Présentation du nouveau cours
- 3 Exercices
- 4 Avec une petite pause pour souffler un peu...

**Examen Session 1 :** 2 heures, avec documents... le 10 décembre 2018

**Examen Session 2 :** 2 heures, avec documents... ???

Des questions ?

## Différents paradigmes de programmation

### Paradigme fonctionnel

```

1  Pgcd(a, b) =
2    Si a = b Alors
3      a
4    SinonSi a > b Alors
5      pgcd(a-b, b)
6    Sinon
7      pgcd(a, b-a)
8    FinSi

```

### Paradigme impératif

```

1  Pgcd(a, b) =
2    TantQue a <> b Faire
3      Si a > b Alors
4        a <- a - b
5      Sinon
6        b <- b - a
7      FinSi
8    FinTQ
9    Résultat <- a

```

### Paradigme déclaratif (logique)

```

1  Pgcd(a, a, a).
2  Pgcd(a, b, r) :- a > b, a1 is a - b, Pgcd(a1, b, r).
3  Pgcd(a, b, r) :- a < b, Pgcd(b, a, r).

```

## Comparaison des différents paradigmes

### Paradigme impératif

- Le programme est une instruction complexe.
- Un programme est composé d'instructions qui modifient l'état du programme
- Ces effets de bord rendent difficile le raisonnement sur un programme, la parallélisation du code (problèmes de synchronisation), etc.

### Paradigme fonctionnel

- Le programme est une fonction.
- Un programme est composé de fonctions (au sens mathématique).
- Pas d'état. Pas d'effet de bord.
  - Raisonnement facilité sur le programme.
  - Mise en œuvre plus facile (voire gratuite) de la concurrence.

### Paradigme déclaratif

- Le programme est un but.
- Un programme est composé d'un ensemble de faits et règles.
- Le programmeur ne donne pas de description opératoire (l'ordre d'application des règles)
  - Concurrence gratuite
  - Raisonnement facilité

**Évolution** : De plus en plus, les langages sont multi-paradigmes !

## Quelques enjeux de la programmation

### Faciliter d'écriture d'un programme

- de petite taille (quelques centaines à milliers de lignes), de grande taille (millions de lignes)
- par un informaticien, par un fonctionnel ?

### Capacité à faire évoluer un programme

- prendre en compte de nouveaux besoins
- prendre en compte les évolutions réglementaires, technologiques, etc.
- identifier et corriger les erreurs...

### Capacité à raisonner sur un programme (pour gagner en confiance dans le programme)

- Est-ce que le programme fait ce qui est attendu ?
- Preuve ? Vérification exhaustive ? Test ? Certification ?

### Avoir des programmes efficaces : bien utiliser les ressources disponibles

- multi-coeurs, multi-processeurs, architectures parallèles, etc.
- espace mémoire, communications réseau, etc.

### Quels sont les enjeux prioritaires ?