

# Analyse et conduite de projets (UE205) – Les design patterns

HELHa - Mons  
Bachelier en Informatique de Gestion  
Bloc 2  
Année académique 2019-2020  
A. Colmant  
L. Godefroid

## Sources

- Design patterns pour Java : les 23 modèles de conception : descriptions et solutions illustrées en UML 2 et Java, Laurent Debrauwer, ENI Editions.
- Design patterns pour Java : Mise en oeuvre des modèles de conception en Java, Exercices et corrigés, Laurent Debrauwer et Naouel Karam, ENI Editions.
- Head first Design patterns, Eric et Elisabeth Freeman, O'Reilly



# Sommaire

- Introduction aux patterns
- Patterns de comportement
- Patterns de construction
- Patterns de structuration

# Introduction aux patterns

- Méthode d'apprentissage
- Définition
- Classification des patterns
- Choix d'un design pattern pour un problème

# Méthode d'apprentissage

- Objectif : comprendre le fonctionnement de différents design patterns et pouvoir les appliquer à l'aide du langage Java.
- Méthode d'apprentissage : présentation théorique des différents design patterns (avec diagramme de classes associé) suivie d'un exemple concret pour chacun d'eux. Exercices pratiques en Java.

## Définition (1)

- Un design pattern consiste en un schéma à objets qui forme une solution à un problème connu et fréquent.
- Les design patterns permettent de répondre à des problèmes de conception de logiciels orientés objets.
- Il n'y a pas de réelle formalisation des patterns, il s'agit de solutions éprouvées par des programmeurs.



## Définition (2)

- Un pattern est indépendant des langages. C'est une solution abstraite de haut niveau.
- Les patterns respectent les bonnes pratiques de programmation OO. Ils s'appuient sur des concepts OO (polymorphisme, méthodes abstraites,...).
- Un pattern s'exprime souvent par des interfaces et des classes abstraites.



# Avantages des DP

- Vocabulaire commun.
- Capitalisation de l'expérience (catalogue de solutions).
- Niveau d'abstraction plus élevé améliorant la construction des logiciels.
- Réduction de la complexité du développement de logiciel.



# Inconvénients des DP

- Effort d'abstraction et de synthèse.
- Apprentissage des patterns.

# Classification des patterns

Il existe une classification des patterns selon leur vocation :

- Les patterns de comportement permettent l'organisation des interactions et la répartition des traitements.
- Les patterns de construction servent à organiser la création d'objets.
- Les patterns de structuration permettent de simplifier la hiérarchie des classes et leurs relations.

# Les DP de base

## Patterns de création

Abstract Factory  
Builder  
FactoryMethod  
Prototype  
Singleton

## Patterns structurels

Adapter  
Bridge  
Composite  
Decorator  
Facade  
Flyweight  
Proxy

## Patterns de comportement

Chaine of responsibility  
Command  
Interpreter  
Iterator  
Mediator  
Memento  
Observer  
State  
Strategy  
Template method  
Visitor



# Choix d'un design pattern pour un problème

- Il faut tout d'abord s'assurer que le design pattern corresponde bien au problème posé et permette de le résoudre efficacement.
- Quand le design pattern approprié est choisi, il faut bien évidemment l'adapter au problème considéré (nom des classes, des méthodes,...).