

# UE 1 : Architecture logicielle

- Cours 6 :
  - Architecture de Framework
  - Exemples

## Références sur les Frameworks

- M.E. Fayad, D.C. Schmidt, R.E. Johnson (1999), Wiley
  - *Building Application Frameworks.*
  - *Implementing Application Frameworks.*
  - *Domain-Specific Application Frameworks.*

## Développement de logiciels à objets à grande échelle

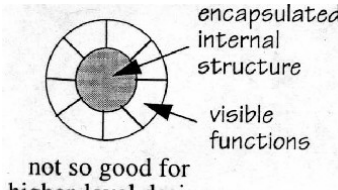
- Un développement plus rapide et efficace grâce à la réutilisation de conception et de code.
- Un framework logiciel (à objets) est une conception réutilisable avec une implémentation
  - Conception : modèle du domaine de l'application
  - Implémentation : comment le modèle peut être exécuté
- Une expérience cumulée pour l'architecture logicielle et son implémentation pour un domaine d'application.
- Adaptation pour résoudre un problème particulier du domaine d'application.

## Définitions (1)

- *Un framework est un ensemble de classes et de types d'interface qui structure les mécanismes essentiels d'un domaine particulier*
- *Un framework est une conception réutilisable de tout ou d'une partie d'un système qui est représentée par un ensemble de classes abstraites et par la façon dont leurs instances interagissent.*
- *Un framework est une application réutilisable, semi-complète qui peut être spécialisée pour produire des applications.[Johnson-Foote 1988]*
- *Un framework est le squelette d'une application qui peut être adaptée par un développeur d'application.*

## Définitions (2)

Image traditionnelle d'un objet



encapsulated internal structure

visible functions

not so good for higher-level design

Framework

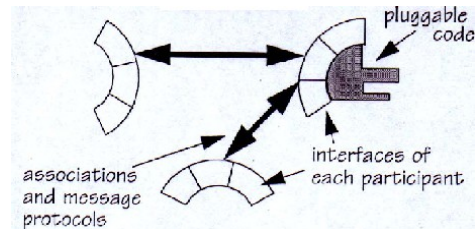


Image empruntée à Alan C.Wills « Frameworks and component-based development »

## Exemples de frameworks

Nom	Usage	Architecture cible	Langage	Auteur
Struts	application web	MVC	java	Apache Software
Hibernate	persistance	Java persistence API	java	JBoss
Spring	développement Java et tests		Java	Apache Software
.Net		ASP.NET	C#, Visual Basic, .Net	Microsoft
Zope	serveur d'application web	Zope	Python	Zope corporation
TopLink	persistance	Java Persistence API	Java	Oracle Corporation
MARF (Modular Audio Recognition Framework)	traitement du signal et de la langue naturelle	pipeline	Java	Serguei Mokhov, Stephen Sinclair

## Frameworks et composants logiciels

- Les frameworks peuvent être vus comme des composants logiciels qui sont conçus pour la réutilisation.
- Un framework
  - offre des composants préfabriqués,
  - capture les éléments de l'architecture et de l'implémentation qui ne varient pas,
  - sépare les parties qui varient de la logique spécifique à l'application (*hot spot/frozen spot*),
  - gère le flot de contrôle.

## Patrons de conception versus Frameworks

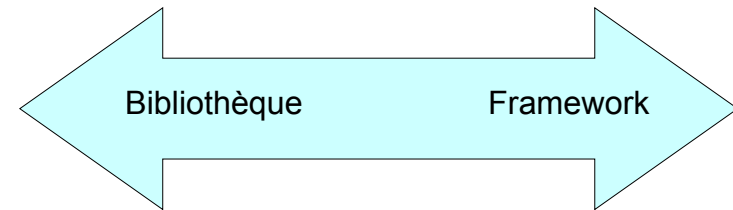
- Frameworks** : réutilisation de conceptions concrètes, d'algorithmes et d'implémentations dans un langage de programmation particulier.
- Un framework logiciel est une mini-architecture réutilisable.
- Patrons de conception** : réutilisation de conceptions abstraites et de petites collections de classes coopérantes.

Les patrons de conception servent de blocs de construction aux frameworks

## Bibliothèques de classes versus frameworks

- Coopération des classes
- Spécificité du domaine
- Classes passives / flot de contrôle
- Utilisation conjointe
- Bibliothèques de classes pour des tâches de base spécifiques

## Différences entre un Framework et une bibliothèque de classes



- Ensemble de classes instanciées par le client
- Les clients appellent les fonctions
- Pas de flot de contrôle prédéfini
- Pas d'interaction prédéfinie
- Pas de comportement par défaut
- Fournit l'adaptation par spécialisation
- Appelle les fonctions clientes
- Contrôle le flot d'exécution
- Définit l'interaction des objets
- Fournit un comportement par défaut

## Composants versus Frameworks

- Les composants :
  - instances de classe
  - boîte noire
  - réutilisation
- Les frameworks :
  - utilisation des composants
  - infrastructure et répartition du logiciel

## Caractéristiques distinctives d'un framework

- Inversion de contrôle
  - Contrôle complètement le flot des programmes
- Comportement par défaut
  - Comportement utile qui peut être redéfini
- Extensibilité
  - Des fonctionnalités
- Code non-modifiable
  - La modification de code n'est pas permise mais il eut être étendu.

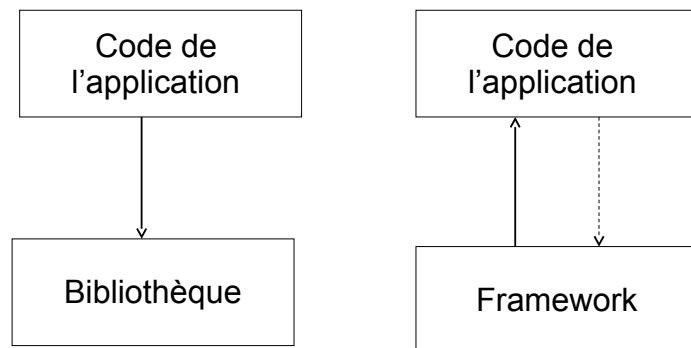
## Avantages et limitations des frameworks

- Le plus gros avantage est la réduction de temps et d'énergie pour développer un logiciel
- Les propriétés :
  - Modularité
  - Réutilisabilité
  - Extensibilité avec les « hook methods »
  - Inversion de contrôle
- Par contre des limitations
  - Temps d'apprentissage pour un débutant
  - coût de développement
  - aspects spécifiques (efficacité / généricité)

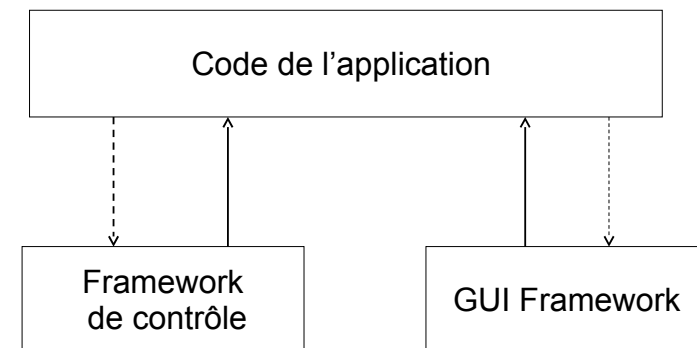
## Inversion du contrôle

- Une caractéristique des frameworks : utilisation extensive de la liaison dynamique.
- L'inversion de contrôle fait partie de l'architecture typique d'un framework.
- Les *frameworks appelants* sont les entités actives dans une application, contrôlant et invoquant les autres parties.
- Les *frameworks appelés* sont les entités passives qui peuvent être appelées par les autres parties de l'application.

## Inversion du contrôle (2)



## Problème de composition de contrôle



## Bénéfices apportés par les frameworks

- Issus de caractéristiques des langages à objets
  - abstraction de données
  - polymorphisme
  - héritage
- Réutilisation
  - interfaces stables
  - composants génériques
- Extensibilité
  - points de variations explicites
- Modularité
  - implémentation/ interfaces stables.
  - Localisation des impacts de changements
- Inversion des rôles

## Maximiser les bénéfices des Frameworks

### **Bénéfices**

- sauvegardes à long terme
- influence l'expérience de l'expert du domaine
- cohérence et meilleure intégration
- réduit la maintenance
- favorise la productivité

### **Coûts**

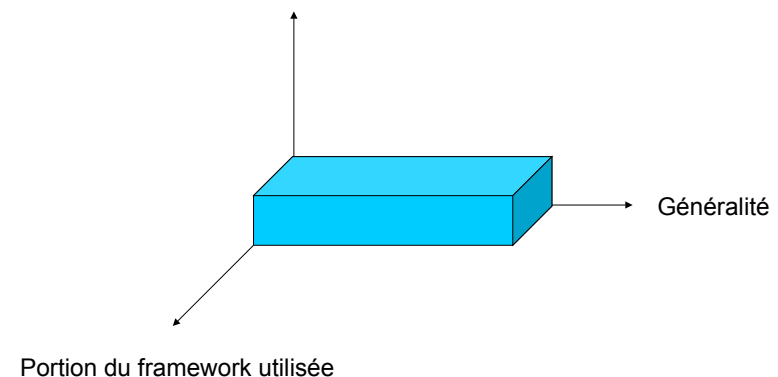
- plus d'effort à construire et à apprendre
- programmes plus difficiles à mettre au point
- documentation, maintenance et support.

## Framework horizontal versus Framework vertical

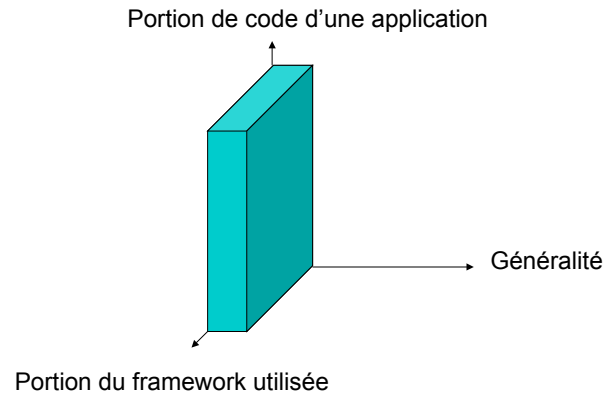
- Un framework horizontal est plus général qu'un vertical et peut être utilisé par plus de types d'applications.
- Les frameworks verticaux sont spécifiques à un domaine d'application particulier.
- Trois variables
  - niveau de généralité
  - portion du framework utilisée par l'application
  - portion moyenne de code dans une application qui est construite à partir du framework.

## Framework horizontal

Portion de code d'une application



## Framework vertical



## Framework boîte blanche/ Framework boîte noire

- Framework *boîte blanche*
  - repose sur l'héritage et la liaison dynamique
  - généralisation des classes des applications individuelles
  - dérivation des classes de base du framework
- Framework *boîte noire*
  - repose sur la composition et la délégation
  - extensible en définissant des interfaces pour les composants qui seront intégrés au framework
- Un framework peut être boîte blanche et boîte noire à la fois : perception de la transparence dépend des aspects autre que du code tels que la documentation et les outils .

## Framework boîte blanche

- Quels sont les effets d'une approche basée sur l'héritage ?
  - Il est nécessaire de comprendre comment les sousclasses et superclasses marchent ensemble.
  - Nous avons accès à la fois aux parties protégées et publics de la classe.
  - Pour fournir une fonctionnalité, on peut occulter des méthodes existantes et implémenter des méthodes abstraites.
  - Nous avons accès aux méthodes parent (en appelant `super.method()`).

## Caractérisation des Frameworks par le domaine de problème

- Framework *d'application*
- Framework *de domaine*
- Framework *de support*

## Classification des Frameworks par la position dans le processus de développement

- *Frameworks d'infrastructure* (système d'exploitation, débogueurs, communication, java Swing)
- *Frameworks d'intergiciel* (Java RMI, WebSphere, implémentation de CORBA)
- *Frameworks d'application d'entreprise* (télécommunications, avioniques, manufacturing, ingénierie financière et les activités de processus d'entreprise)

## Liste de frameworks java vs environnement de développement (voir Wikipedia)

- Apache Struts — de la fondation Apache Software
- Eclipse — open-source de la fondation Eclipse Software Foundation
- Hibernate
- JavaServer Faces
- JBoss Seam — de la société JBoss
- Leonardi
- Modular Audio Recognition Framework - open-source du MARF Research and Development Group
- NetBeans — de la société Sun Microsystems
- Nuxeo - Framework open source dédié principalement à la gestion de contenus
- Play - Framework opensource (licence Apache) développé par la société zenexity
- Spring framework

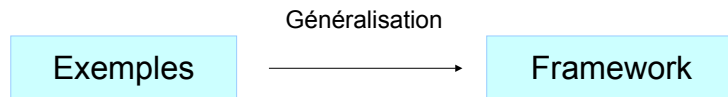
## Structures des Frameworks

- **Manager-driven**
  - Une fonction unique de contrôle déclenche la plupart des actions
- **Architecture-driven**
  - repose sur l'héritage pour l'adaptation.
- **Data-driven**
  - repose sur la composition d'objets pour l'adaptation.

## Comment concevoir un framework?

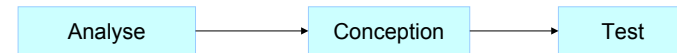
- Référence : tutorial de Ralph Johnson (OOPSLA'93)
- Framework
  - le résultat d'une analyse de domaine
  - décrit comment décomposer le problème
  - est représenté par un programme
- Les abstractions sont trouvées par une analyse ascendante, en examinant des exemples concrets
- Généralisation

## Conception guidée par l'exemple



- La généralisation est itérative
- Pour généraliser plus vite il faut
  - obtenir différents points de vue
  - expliquer la conception courante

## Façon idéale de développer un framework



- Analyser le domaine de problème
  - Apprendre les abstractions connues
  - Récupérer des exemples de programmes
- Concevoir des abstractions qui peuvent être spécialisées pour couvrir les exemples
- Test du framework en l'utilisant pour résoudre les exemples

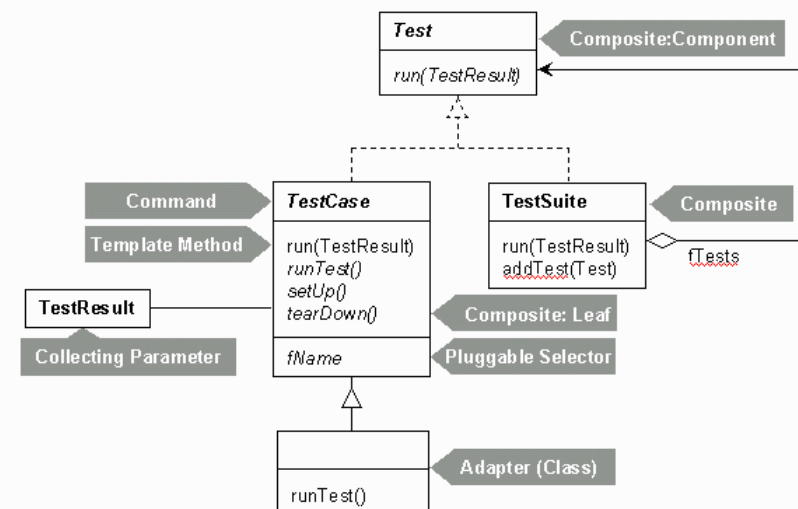
## Critère pour choisir un framework

Le développement d'un framework est un investissement long, coûteux et risqué.  
Pour choisir un framework du marché il faut prendre en compte plusieurs caractéristiques :

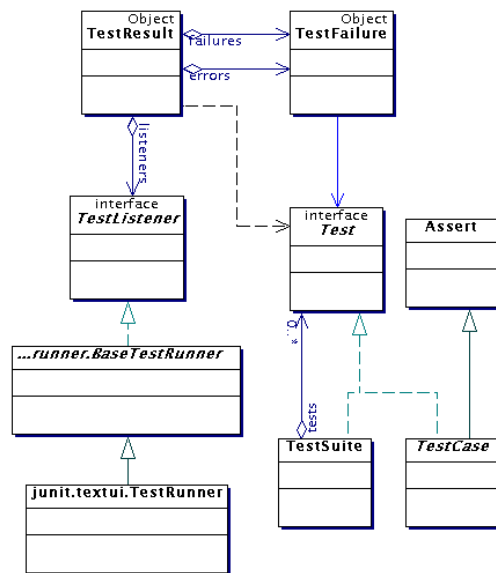
- Adoption par la communauté
- La qualité de la documentation
- Le support (commercial ou communautaire)
- Le support par les outils de développements

Attention à la complexité, au temps d'apprentissage pour avoir la maîtrise et être efficace.

## Extrait du framework Junit







# Java Collections Framework

(from java tutorial)

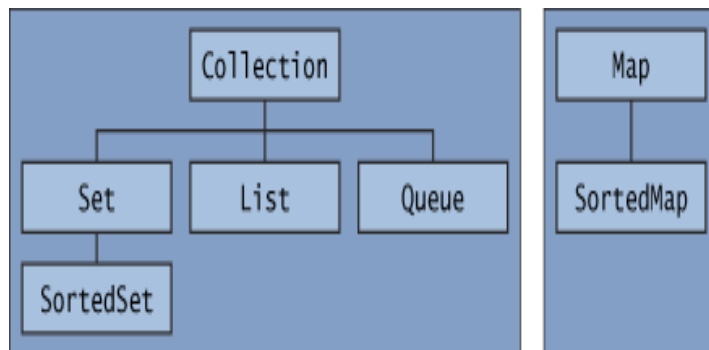
A collections framework is a unified architecture for representing and manipulating collections. All collections frameworks contain the following:

**Interfaces:** These are abstract data types that represent collections. Interfaces allow collections to be manipulated independently of the details of their representation. In object-oriented languages, interfaces generally form a hierarchy.

**Implementations:** These are the concrete implementations of the collection interfaces. In essence, they are reusable data structures.

**Algorithms:** These are the methods that perform useful computations, such as searching and sorting, on objects that implement collection interfaces. The algorithms are said to be polymorphic: that is, the same method can be used on many different implementations of the appropriate collection interface. In essence, algorithms are reusable functionality.

## Collection Interfaces



## General purpose implementations

Interfaces

Implementations

	Hash table	Rezizable array	Tree	Linked list	Hash table + Linked list
Set	HashSet		TreeSet		LinkedHashSet
List		ArrayList		LinkedList	
Queue					
Map	HashMap		TreeMap		LinkedHashMap

## Plus d'exemples

- dans M.E. Fayad, D.C. Schmidt, R.E. Johnson, *Domain-Specific Application Frameworks*, Wiley, 1999.
  - Computer-Integrated Manufacturing Frameworks
  - Distributed-system frameworks
  - Network and Telecommunication Frameworks

## Les Applets Java : a simple Framework (1)

- Les applets Java sont des programmes java qui s'exécutent dans un navigateur Web
- le package java.applet est un framework d'application simple; il contient des superclasses pour faire des applets.
- Pour implémenter une applet, on écrit sa classe comme dérivant de la classe Applet.

## Le package java.applet

<http://java.sun.com/javase/7/docs/api/>

- Fournit les classes nécessaires pour créer une applet et les classes qu'une applet utilise pour communiquer avec son contexte applet.
- Le framework `applet` comprend deux entités: `applet` et `applet context`. Une applet est une fenêtre enfichable (sous-classe de Panel) avec quelques méthodes en plus que le contexte de l'applet peut utiliser pour initialiser, démarrer et stopper l'applet.
- `applet context` est une application qui est responsable du chargement et de l'exécution des applets. Par exemple, l'applet context pourrait être un navigateur Web ou un environnement de développement d'applet.

- Le package fournit les interfaces : `AppletContext`, `AppletStub` et `AudioClip`, et la classe `Applet`.

### Interface Summary

Interface	Description
<code>AppletContext</code>	This interface corresponds to an applet's environment: the document containing the applet and the other applets in the same document.
<code>AppletStub</code>	When an applet is first created, an applet stub is attached to it using the applet's <code>setStub</code> method.
<code>AudioClip</code>	The <code>AudioClip</code> interface is a simple abstraction for playing a sound clip.

## Les Applets Java : a simple Framework (2)

- On doit surcharger certaines des méthodes suivantes :
  - init – initialise les données
  - start – démarre les threads de l'applet
  - stop – arrête les threads
  - destroy – libère les ressources
- La méthode principale est fournie par le framework et non par le programmeur de l'application.
- Le séquençement des méthodes fournies par les programmeurs sont sous le contrôle du framework.