Rappel des principes OO - 1

- Les requêtes sont les seuls moyens de faire exécuter une opération à un objet.
- Les opérations sont les seuls moyens de modifier les données internes d'un objet.
- L'état interne d'un objet est dit **encapsulé**

Rappel des principes OO - 2

- L'inetrface d'un objet est constituée de l'ensemble des signatures de sa classe.
- Les objets ne sont connus qu'au travers de leurs interfaces.
- L'interface ne révèle rien de la réalisation.
- 2 objets dotés d'implémentation différentes peuvent avoir des interfaces identiques.

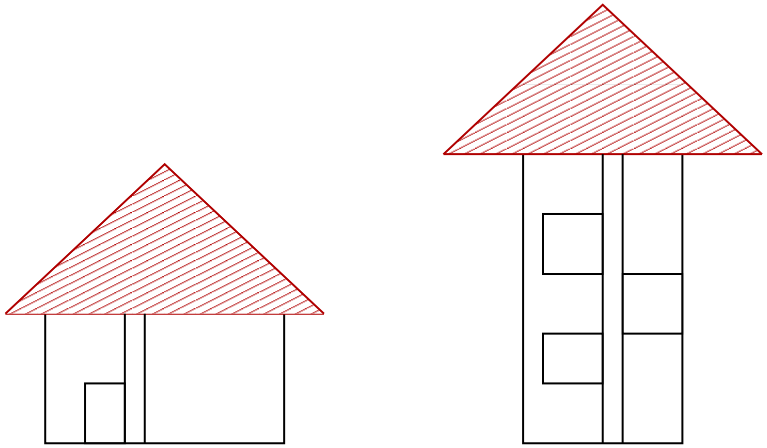
Rappel des principes OO - 3

- L’envoi d’une requête dépend de la requête et de l’objet recepteur.
- Des objets différents qui répondent à des requêtes identiques peuvent avoir des codes différents.
- L’association à l’exécution d’une requête à un objet définie une liaison dynamique.
- La liaison dynamique fait que le lancement d’une requête n’impose pas d’avoir à se conformer à une implémentation particulière jusqu’à l’exécution.
- Permet la substitution d’objet à l ’exécution si ils possèdent la même interface. Ceci correspond au **polymorphisme**.

Introduction aux Design Patterns

C'est un architecte C. Alexander [a] qui a développé l'idée d'identifier des objets apportant :

"Une solution précise à problème donné, dans un context précis"



- Définir un pattern, c'est fournir le *pourquoi* et le *comment* de chaque solution.

[a] "Timeless way of building", 1979.

Spécificité des “Design Patterns” a

Introduction du concepts de patterns dans la modélisation objet.

- Décrire aussi les relations avec les autres objets du modèle et les conséquences liées à leur utilisation.
- Les patterns modélisent des ***objets qui n’existent pas en tant qu’entités dans le système cognitif humain***.

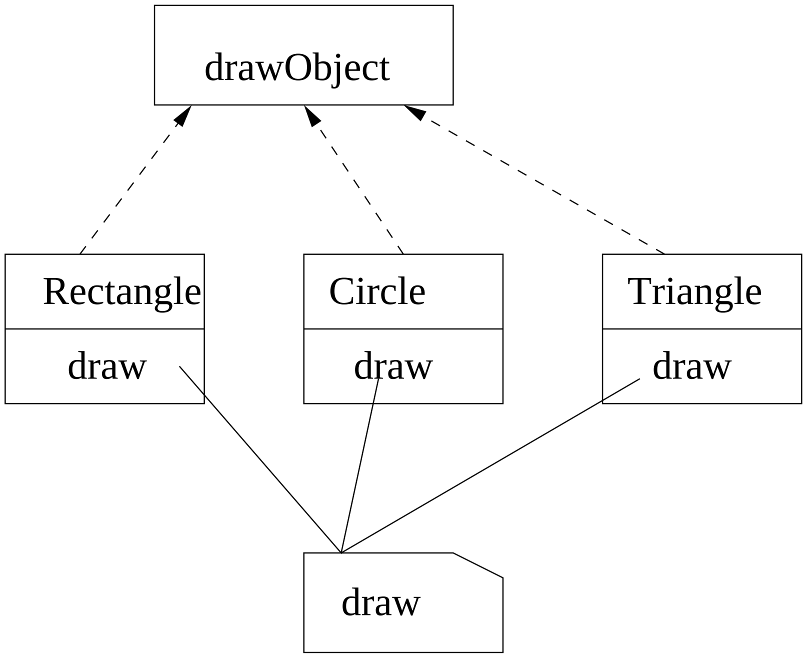
Le catalogue

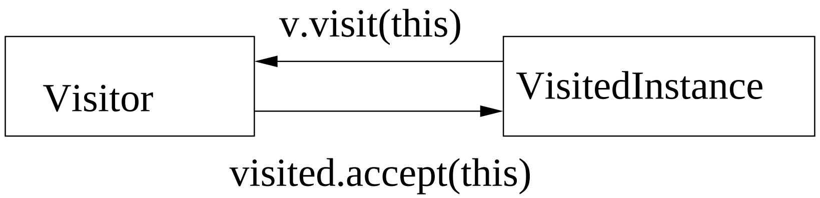
- 23 design patterns classifiés en 3 groupes d'après leur **rôle**

Création	**Structuration**	**Def. de Comportement**
AbstractFactory	Adapter, Bridge	Interpreter, Command
Factory	Composite,	Chain of Responsability
Builder	Facade, Proxy	Iterator, Mediator,Template
Prototype	Flyweight	Memento, Observer
Singleton	Decorator	State, Strategy, Visitor

Un exemple : Visitor

- Objectif : partager du code entre plusieurs objets d’une structure sans lien d’héritage.





public void accept(Vistor v) {v.visit(this);}