HELHa
Haute École Louvain
en Hainaut

Patterns de comportement

- Introduction
- State
- Iterator
- Observer
- Strategy
- Template Method

HELHa
Haute École Louvain en Hainaut

Introduction

- Les patterns de comportement ont pour but de fournir des solutions pour distribuer les traitements et les algorithmes entre objets.
- Ils vont donc spécifier les flux de contrôle et de traitement entre les objets.
- La distribution des traitements pourra se faire soit par héritage (méthodes abstraites définies dans les sous-classes) soit par délégation (coopération entre des objets pour réaliser un traitement).

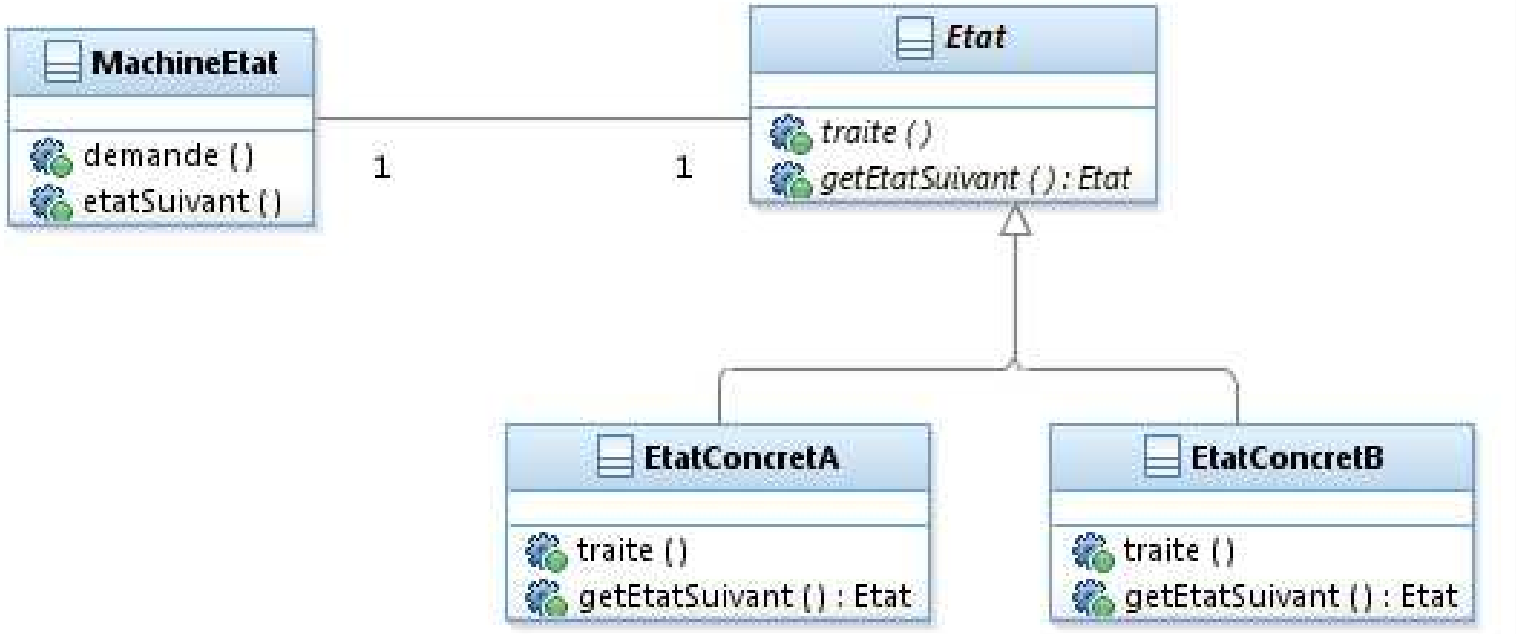
HELHa
Haute École Louvain
en Hainaut

State - utilité

- Permet d'adapter les comportements d'un objet en fonction de son état.
- Utile lorsqu'un objet possède plusieurs états possibles et que les traitements diffèrent dans chaque état.
- Distribution des traitements par délégation.

HELHa
Haute École Louvain
en Hainaut

State - UML



HELHa
Haute École Louvain en Hainaut

State – explications UML

- MachineEtat représente une classe décrivant des objets qui possèdent différents états (qui peuvent donc être représentés dans un diagramme de machine à état).
- Etat est une classe abstraite introduisant les signatures des méthodes (méthodes abstraites) relatives aux différents états. Elle gère également l'association avec MachineEtat.
- EtatConcretA et EtatConcretB sont les sous-classes concrètes qui définissent les comportements introduits par Etat.

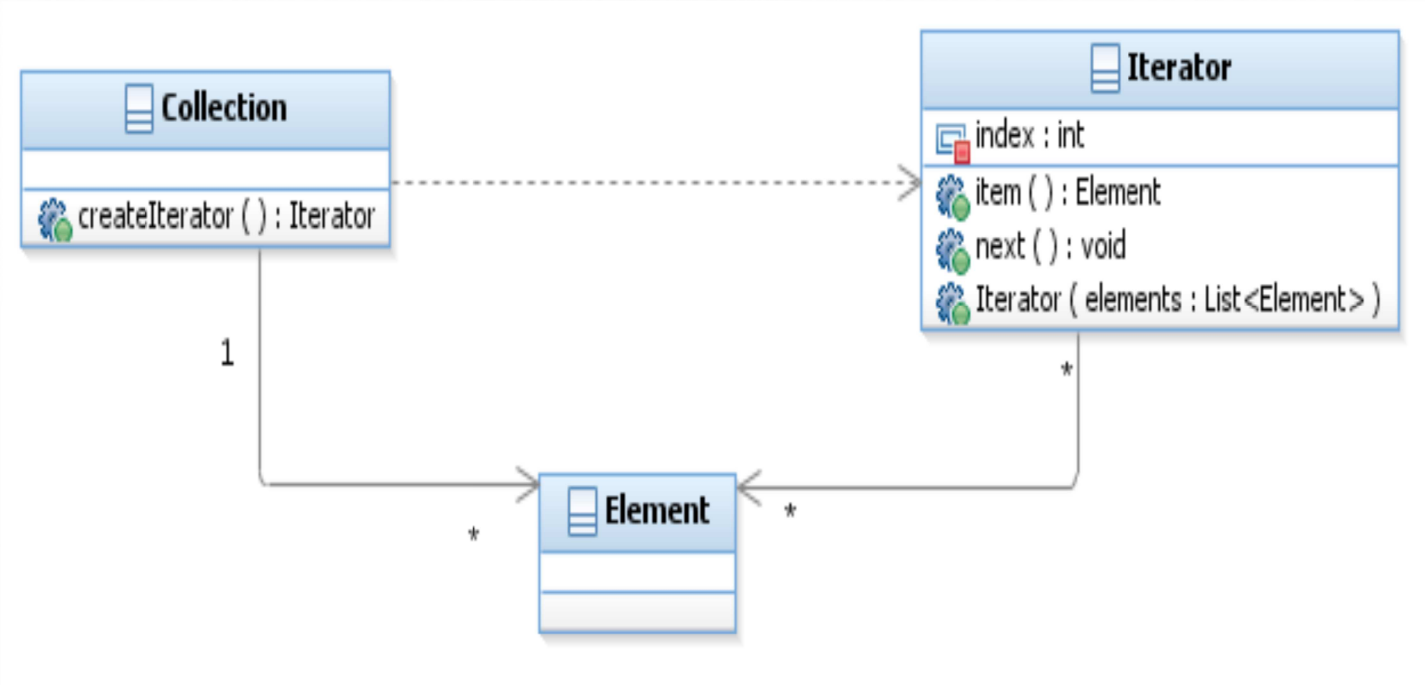
HELHa
Haute École Louvain
en Hainaut

Iterator - utilité

- Permet d'accéder de manière séquentielle à une collection d'objets sans se préoccuper de l'implantation de la collection.
- Rend possible la gestion de plusieurs parcours simultanés d'une collection.
- Distribution des traitements par délégation.

HELHa
Haute École Louvain
en Hainaut

Iterator – UML



HELHa
Haute École Louvain
en Hainaut

Iterator – explications UML

- Collection est une classe qui crée l'association de la collection avec ses éléments et définit la méthode de création de l'itérateur.
- Iterator est une classe qui crée l'association avec les éléments de la collection et qui permet de parcourir la liste des éléments.
- Element est la classe des éléments de la collection.

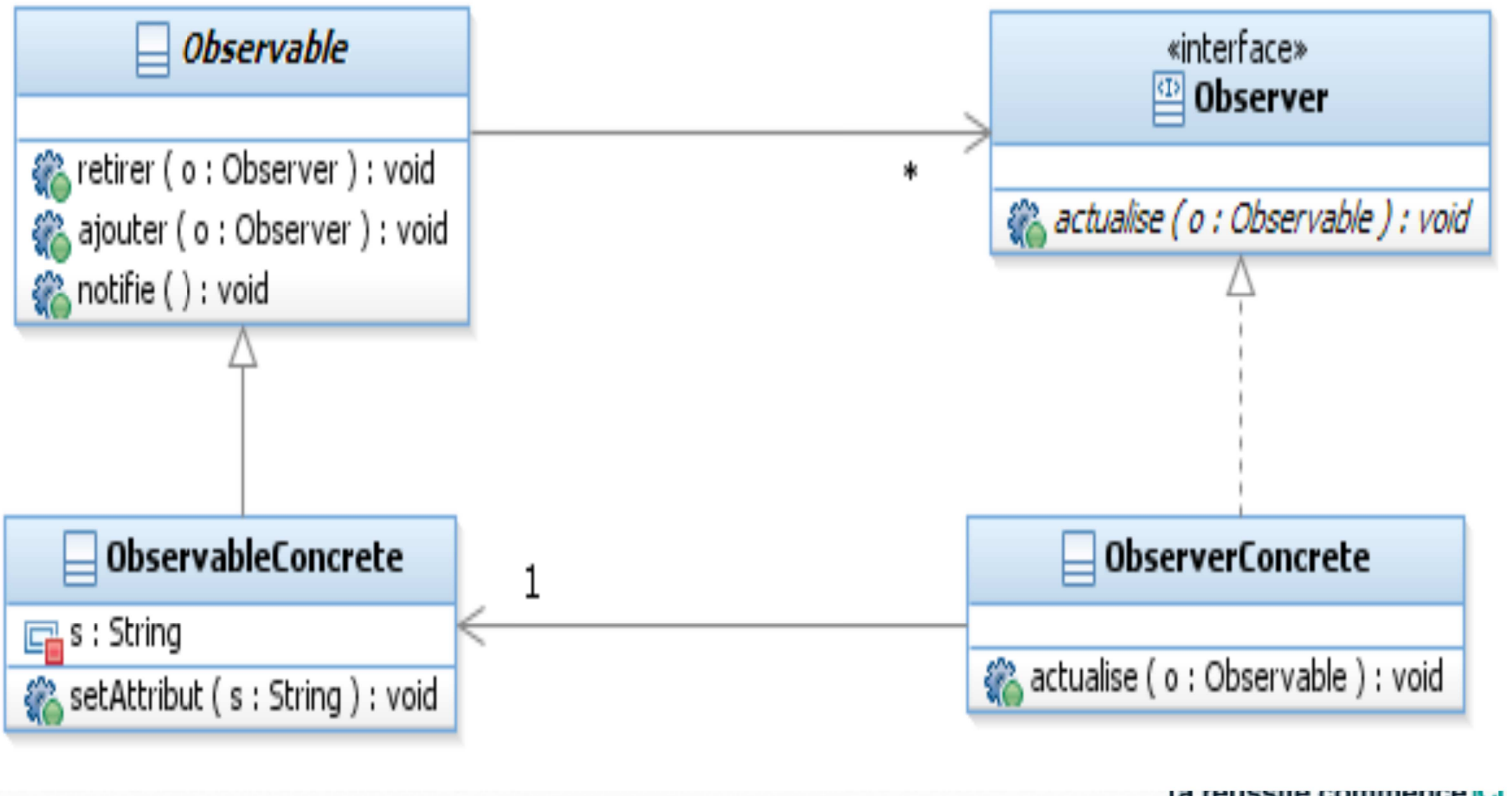
HELHa
Haute École Louvain
en Hainaut

Observer - utilité

- Permet de prévenir des objets observateurs du changement d'état d'un objet observé.
- L'objet observé ne doit pas connaître le type des objets observateurs.
- Distribution des traitements par délégation.

HELHa
Haute École Louvain
en Hainaut

Observer - UML



HELHa
Haute École Louvain en Hainaut

Observer – explications UML

- Observer est une interface définissant une méthode permettant de recevoir les notifications de l'objet observé.
- ObserverConcrete donne une définition à la méthode recevant la notification. Elle peut demander des informations à l'objet observé pour réaliser sa mise à jour.
- Observable est une classe abstraite qui introduit l'association avec les objets observateurs.
- ObservableConcret envoie la notification quand il change d'état.

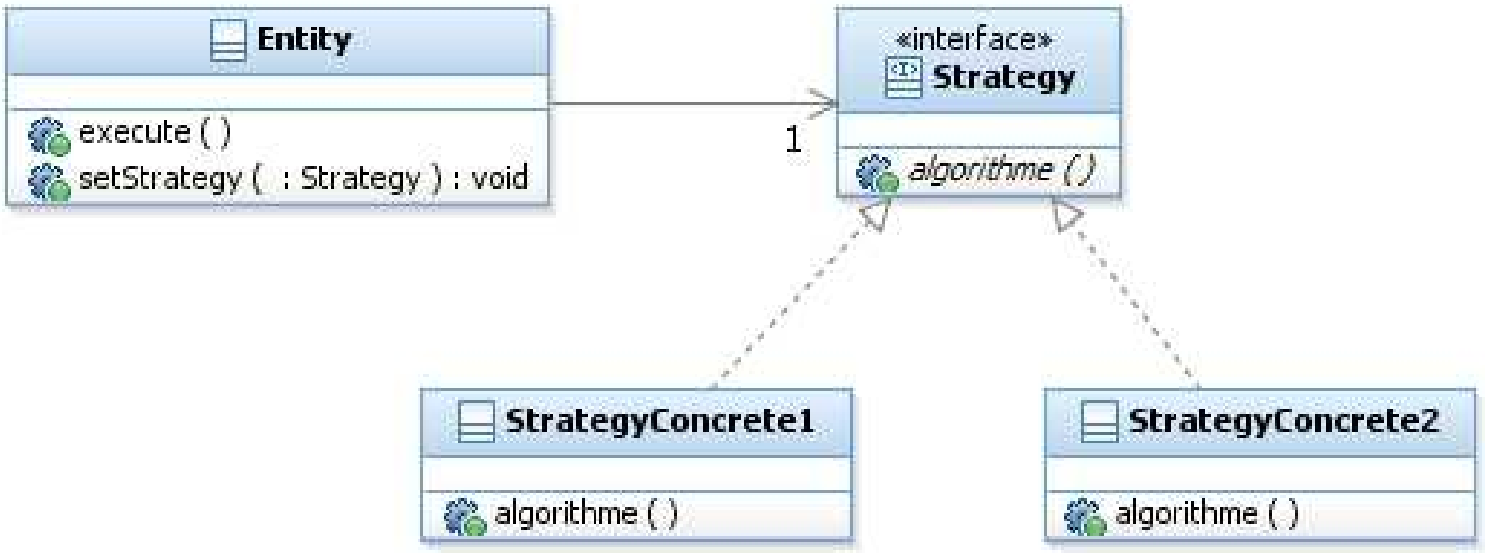
HELHa
Haute École Louvain en Hainaut

Strategy - utilité

- Permet d'adapter le comportement et les algorithmes d'un objet en fonction d'un besoin.
- Évite les algorithmes avec des conditions trop complexes.
- Permet de ne pas réécrire des classes qui ne diffèrent que par une partie de leur comportement.
- Distribution des traitements par délégation.

HELHa
Haute École Louvain
en Hainaut

Strategy - UML



HELHa
Haute École Louvain
en Hainaut

Strategy – explications UML

- Entity est la classe qui utilise un algorithme de Strategy pour réaliser un de ses comportements. Elle peut changer de stratégie grâce à setStrategy.
- Strategy est une interface définissant l'en-tête de la méthode implantant le comportement souhaité.
- StrategyConcrete donnent une définition au comportement défini dans l'interface.

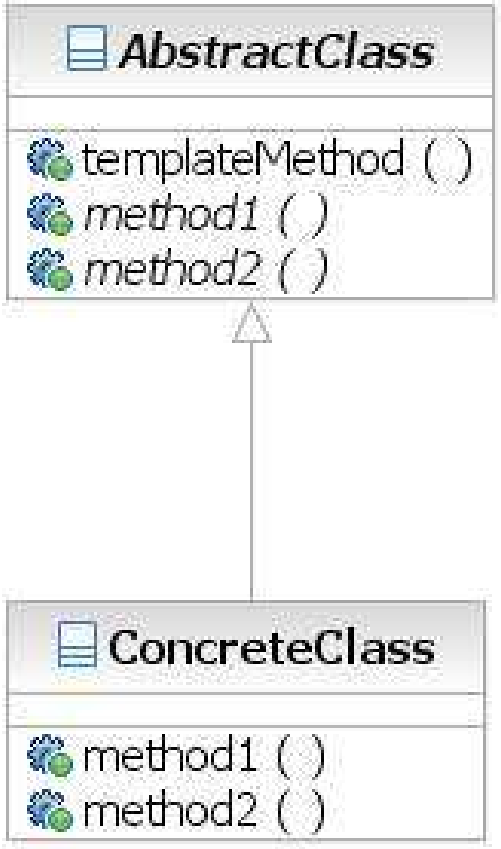
HELHa
Haute École Louvain
en Hainaut

Template Method – utilité

- Permet de laisser aux sous-classes une partie de la réalisation d'une opération.
- Sert à factoriser du code identique entre plusieurs classes en laissant les spécificités dans les sous-classes.
- Distribution des traitements par héritage.

HELHa
Haute École Louvain
en Hainaut

Template Method - UML



HELHa
Haute École Louvain
en Hainaut

Template Method – explications UML

- AbstractClass définit la méthode de base ainsi que l'en-tête des méthodes invoquées dans celle-ci.
- ConcreteClass définit les méthodes abstraites utilisées dans AbstractClass.