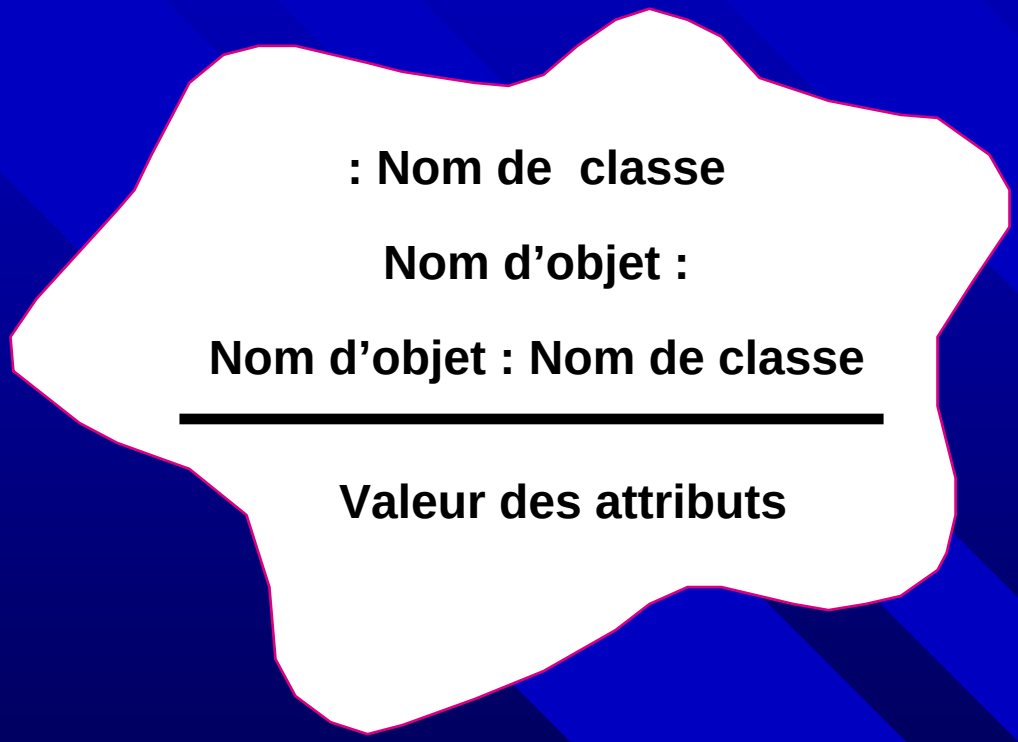
Les objets et les classes

- Les objets
- Les collaborations entre objets
- Les classes
- Les contrats de classes

Les objets

- Les objets du monde réel nous entourent, ils naissent, vivent et meurent
- Les objets informatiques définissent une représentation simplifiée des entités du monde réel
- Les objets représentent des entités concrètes (avec une masse) ou abstraites (concept)
- Les objets encapsulent une partie de la connaissance du monde dans lequel ils évoluent

Représentation graphique des objets



Les objets informatiques sont des abstractions

- Une abstraction est un résumé, un condensé
- Mise en avant des caractéristiques essentielles
- Dissimulation des détails
- Une abstraction se définit par rapport à un point de vue

Exemples d'abstractions

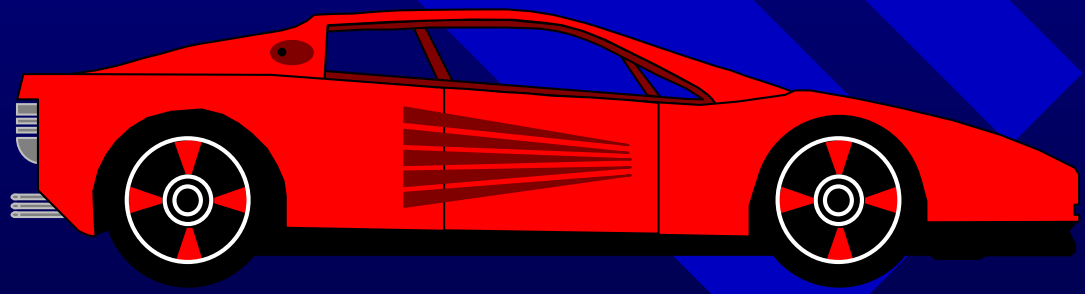
- Une carte routière
- Un nombre complexe
- Un téléviseur
- Une transaction bancaire
- Une porte logique
- Une pile

Caractéristiques fondamentales d'un objet

- L'état
- Le comportement
- L'identité
- La simultanéité
- Le synchronisme
- La persistance
- Le rôle

L’état

- L’état regroupe les valeurs instantanées de tous les attributs d’un objet
- L’état évolue au cours du temps
- L’état instantané d’un objet est la conséquence de ses comportements passés
- Pour un signal électrique : l’amplitude, la pulsation, la phase, ...
- Pour une voiture : la marque, la puissance, la couleur, le nombre de places assises, ...



Le comportement

- Le comportement décrit les réactions d'un objet en termes des modifications de son état et de son interaction avec les autres objets
- Le comportement regroupe toutes les compétences d'un objet
- Un objet peut faire appel aux compétences d'un autre objet
- L'état et le comportement sont liés
  - Le comportement dépend de l'état
  - L'état est modifié par le comportement
- Chaque compétence est appelée méthode ou opération suivant le langage de programmation

L’identité

- En plus de son état un objet possède une identité
- L’identité permet de distinguer tout objet de façon non ambigüe, indépendamment de l’état
- Exemple : notre numéro de sécurité sociale
- Les langages objets utilisent généralement des pointeurs pour réaliser un identifiant
- Une clé primaire dans une base de donnée relationnelle est une manière de réaliser l’identité (en l’insérant dans l’état)

La simultanéité

- Un objet passif ne possède pas de flot d'exécution propre
- Un objet actif possède son propre flot d'exécution
- L'approche orientée-objets unifie les abstractions du monde réel et les abstractions de processus au sein d'un concept unique : l'objet actif
- Une architecture orientée-objets est basée sur une ensemble d'objets collaborants (actifs et passifs)

Le synchronisme

- Un **objet séquentiel** est un objet passif dont la sémantique est préservée uniquement en présence d'un seul flot d'exécution
- Un **objet gardé** est un objet passif dont la sémantique est préservée en présence de plusieurs flots d'exécution
- Un **objet concourant** est un objet actif dont la sémantique est préservée en présence de plusieurs flots d'exécution

La persistance

- Passivation / Activation
- L'existence d'un objet transcende le temps et l'espace
- Sauvegarde de l'état et de la classe d'un objet
- Pas de support de la part des langages de programmation
- Bases de données orientées-objets
- Objets transitoires
- Objets persistants

Le rôle

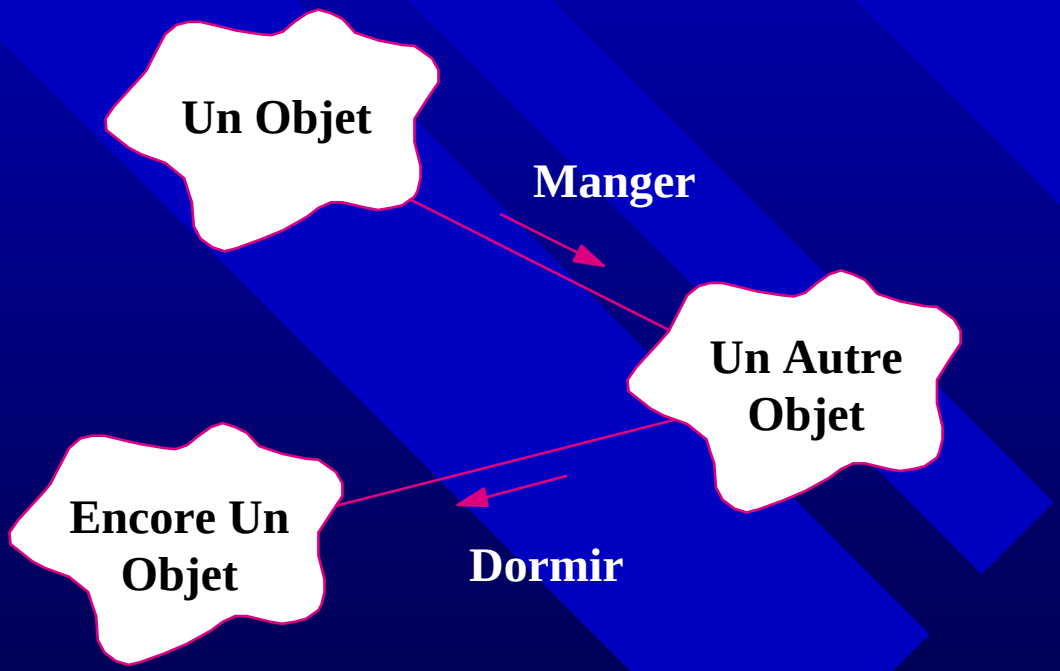
- Un acteur est un objet qui est toujours à l'origine d'une inter-action (un acteur est un objet actif)
- Un serveur est un objet qui n'est jamais à l'origine d'une inter-action
- Un agent est à la fois un acteur et un serveur

Les objets ne vivent pas en ermites

- Les objets inter-agissent les uns avec les autres pour réaliser les fonctionnalités des applications
- Les objets font appel aux compétences de leurs congénères en suivant les contrats spécifiés dans chaque classe
- Lors de l'exécution d'un programme, les objets contribuent solidairement, dans un effort intégré, au bon déroulement de l'application informatique
- Le comportement global d'une application repose donc sur la communication entre les objets qui la composent

Communication entre objets

- Les objets communiquent en échangeant des messages



Le concept de messages

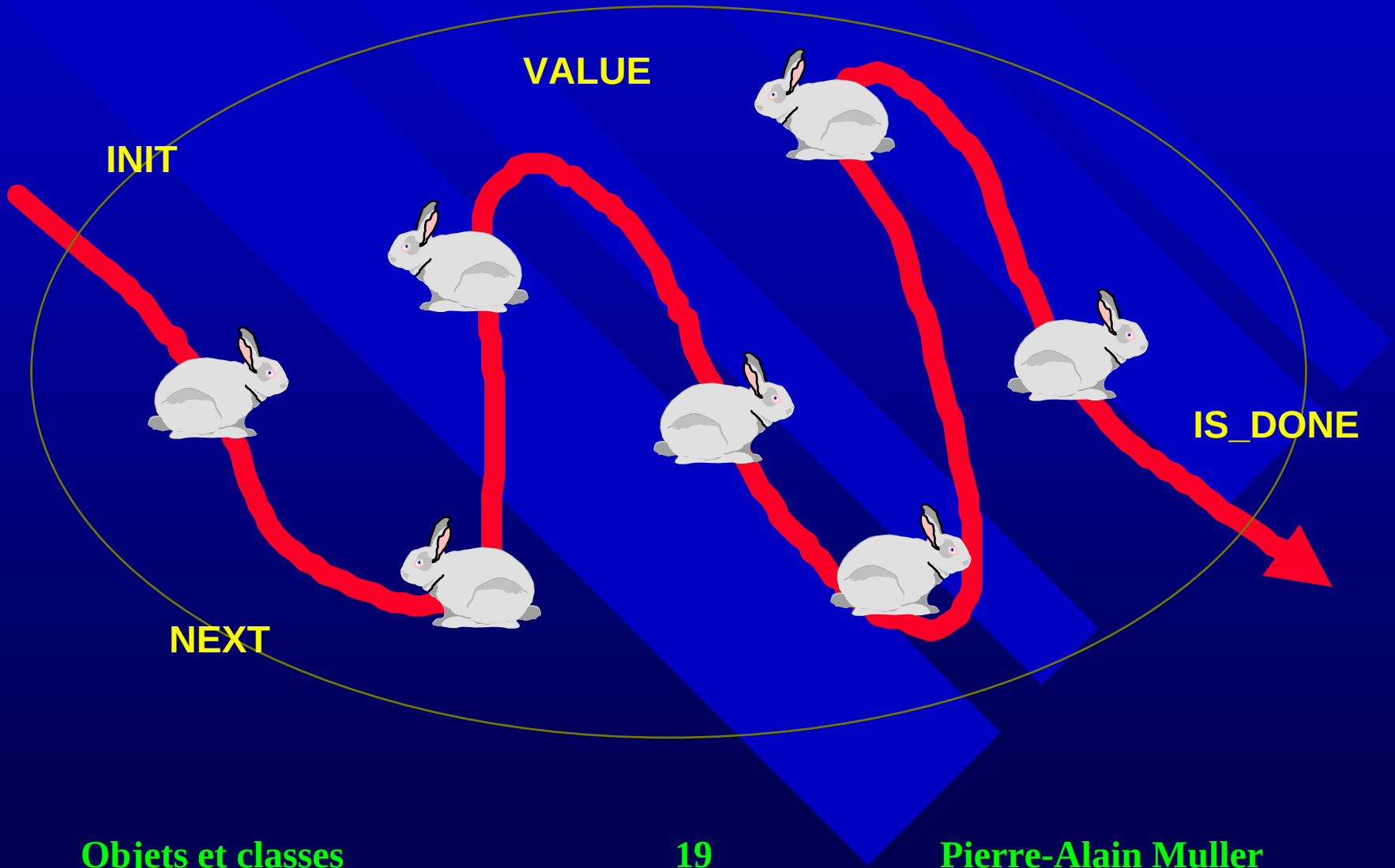
- L’unité de communication entre objets
- Concept très général pouvant être mis en oeuvre suivant de nombreuses variantes
- Regroupe les flots de contrôle et les flots de données
- Représente également les événements

Exemple de diagramme d'objet

Catégories de messages

- Les **constructeurs** qui créent des objets (appartiennent à la métaclasse)
- Les **destructeurs** qui détruisent des objets (appartiennent à la métaclasse)
- Les **sélecteurs** qui renvoient tout ou partie de l'état
- Les **modifieurs** qui changent tout ou partie de l'état
- Les **itérateurs** qui traversent une collection d'objets (appartiennent à la métaclasse)
  - Actifs (l'itération est sous le contrôle direct de l'utilisateur)
  - Passifs (l'itération est sous le contrôle de la collection)

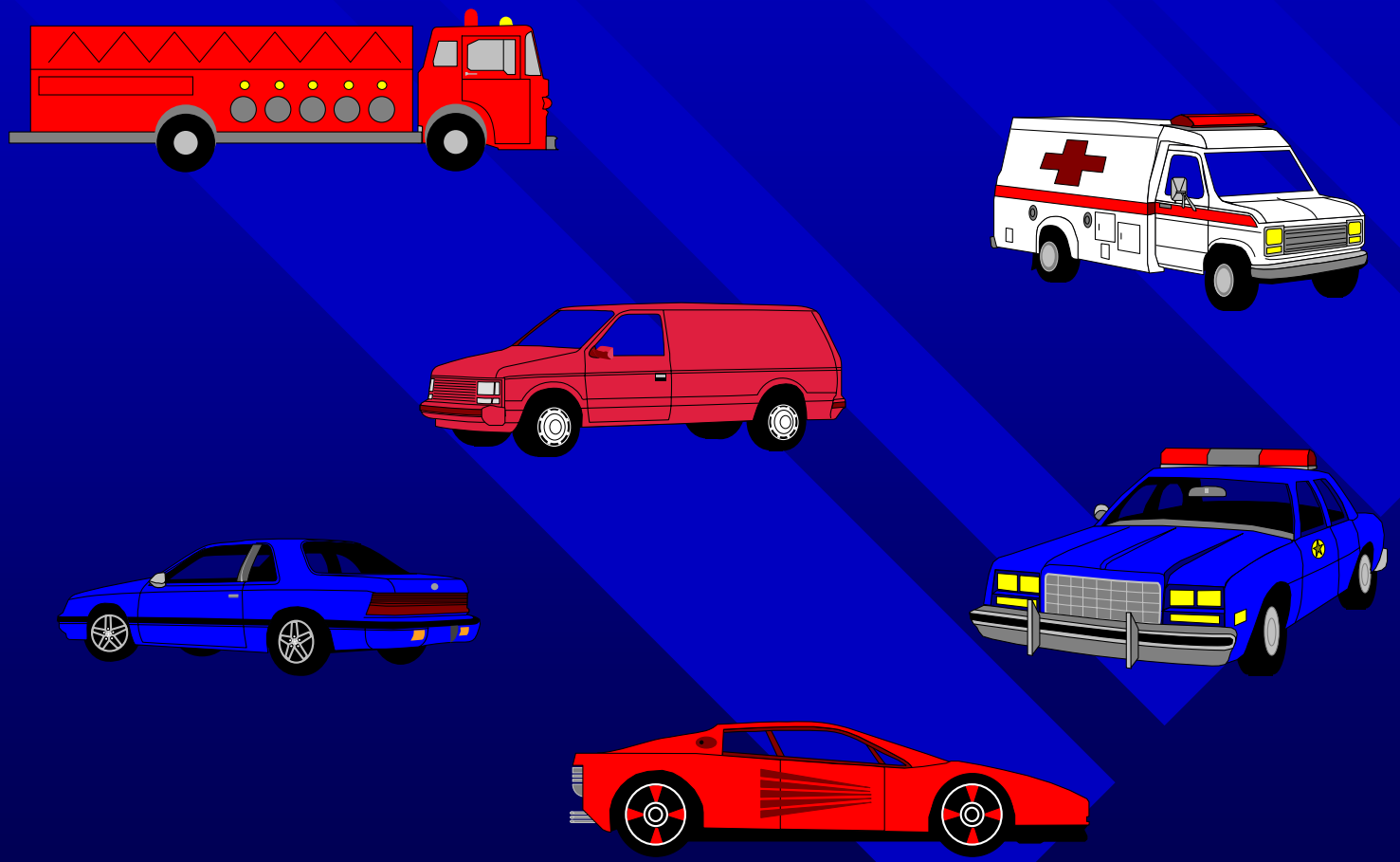
Exemple d’ itérateurs actifs



Le chaos des objets

- Le monde qui nous entoure est constitué de très nombreux objets
- Pour comprendre le monde, l'être humain a tendance à regrouper les éléments qui se ressemblent
- Regrouper des objets suivants des critères de ressemblance s'appelle classer
- Les humains ont classé les animaux, les plantes, les champignons, les atomes, ...

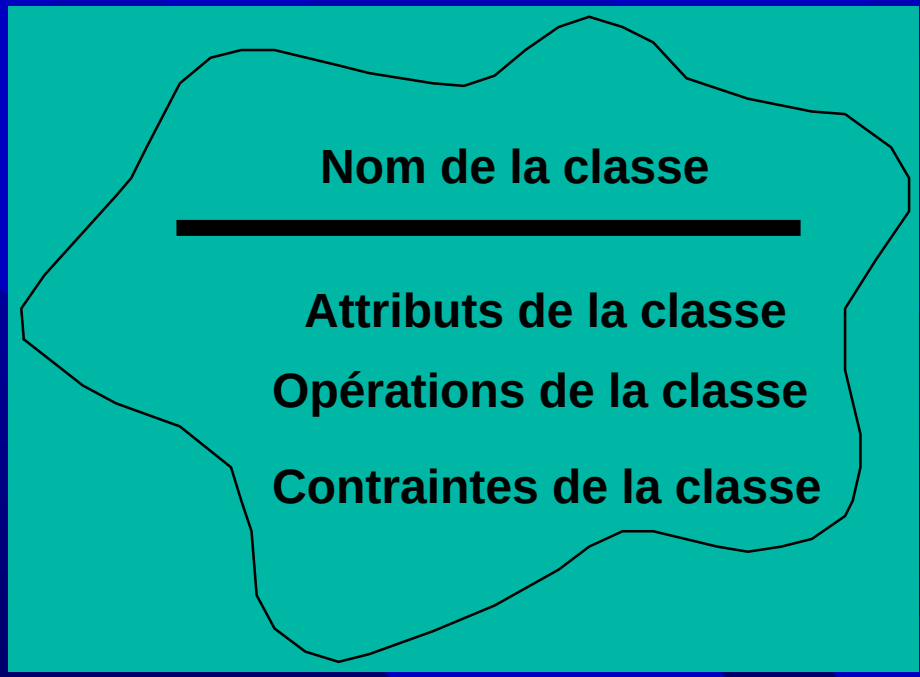
Le chaos des objets (suite)



Les classes

- La classe est une description abstraite d'un ensemble d'objets qui partagent un état, un comportement et des contraintes
- La classe peut être vue comme la factorisation des élements communs à un ensemble d'objets
- La classe décrit le domaine de définition d'un ensemble d'objets (produit cartésien des attributs)
- Certains attributs et opérations peuvent appartenir à la classe seule (pas aux objets) -> métaclasse

Représentation graphique des classes



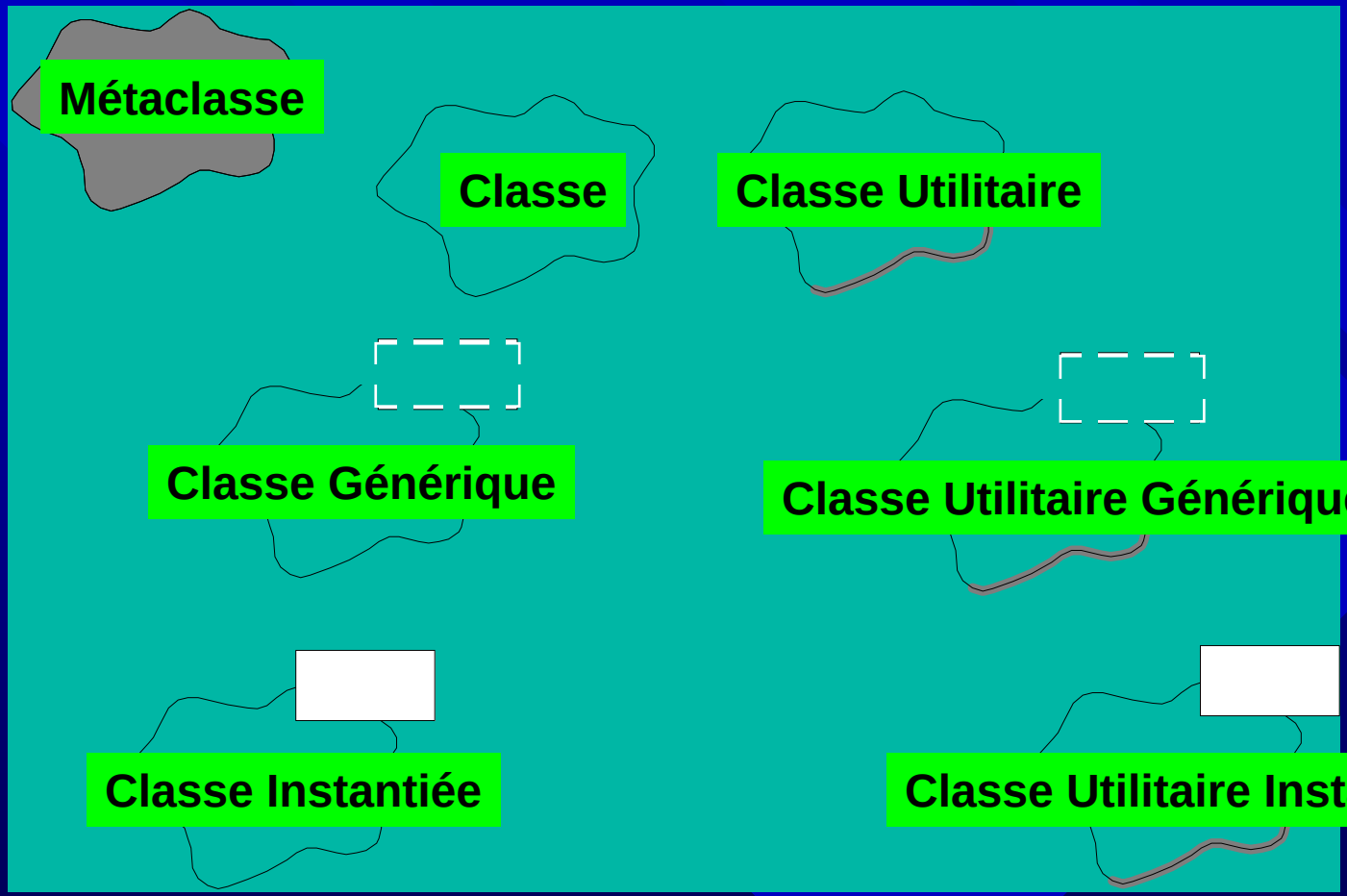
Classe, type de donnée et module

- La classe est une extension du concept de type de donnée abstrait
- La classe sert de moule pour les objets
- Une classe est à la fois un type et un module
- La classe sert de conteneur pour regrouper des données et des opérations

Variantes de classes

- Dans certains cas la classe est réduite à un module
- En Ada une classe est obtenue en plaçant un type privé dans un module
- En C++ il y a un seul type par classe
- En Ada il peut y avoir plusieurs types privés par modules

Différents genres de classes



Le concept de contrat

- Chaque objet encapsule des données dans son état
- Le genre des données encapsulées et les opérations applicables à un objet (donc indirectement aux données encapsulées) sont décrites dans la classe
- Les classes permettent donc de décrire des contrats qui seront valables pour tous les objets issus de cette classe

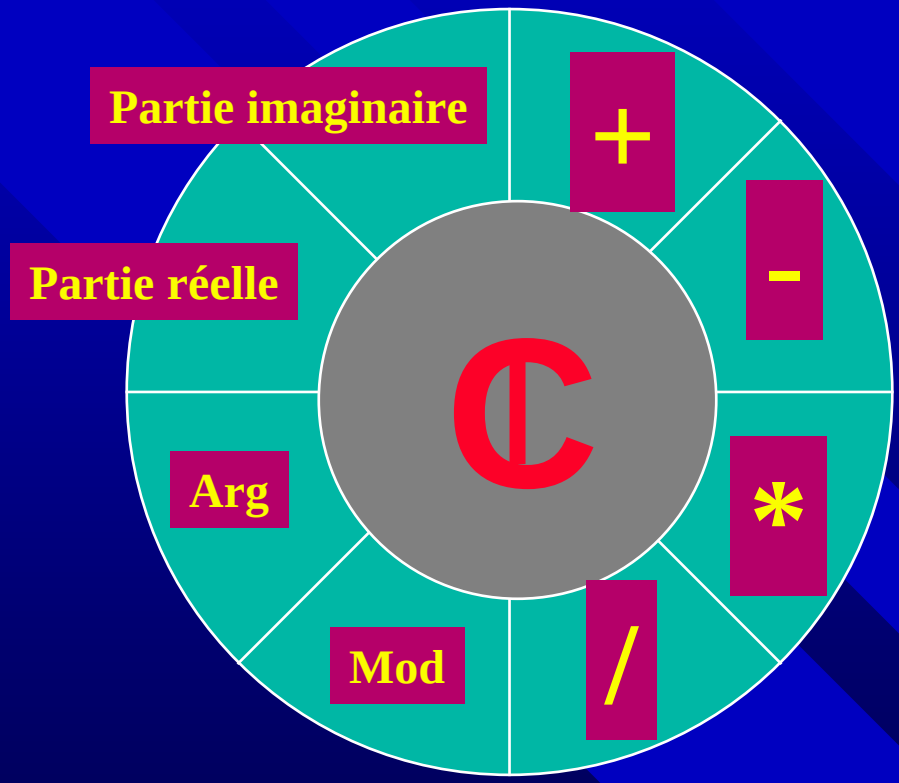
Le concept de contrat (Suite)

- Le contrat d'une classe définit la partie visible des objets, le reste est caché dans l'objet
- Un contrat de classe définit les attributs visibles, les opérations applicables et les contraintes sur les objets
- Ce concept s'appelle l'encapsulation (a priori des données)

Bénéfices de l’encapsulation

- L’encapsulation présente deux avantages
- Les données encapsulées sont protégées des accès intempestifs, ce qui permet de garantir leur intégrité
- Les clients d’une abstraction ne dépendent pas de la réalisation de l’abstraction, mais seulement de son contrat
- Le contenu d’une classe doit être fortement cohérent

Exemple d'encapsulation



Spécification et réalisation des classes

- Le contrat d'une classe est exprimé dans la spécification de la classe (ce que la classe est)
- Comment ce contrat est rempli, c'est-à-dire les choix de représentation, reste caché dans la réalisation de la classe
- Les langages de programmation supportent plus ou moins bien la séparation entre spécification et réalisation
- Ada, C++, Eiffel, Smalltalk, ObjectPascal, ...