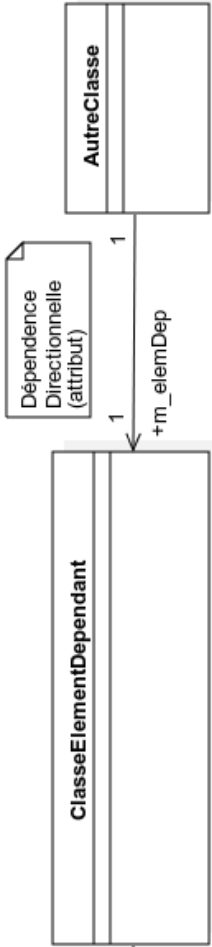


Conventions UML

Diagrammes de Classes

Rémy Malgouyres, Université d'Auvergne



La relation entre ClasseEnsemble et ClasseElementDependant est une composition

- L'élément fait partie d'un seul ensemble
- Si l'ensemble est détruit, l'élément est forcément détruit (cycles de vie liés)

Exemple de relation de composition :

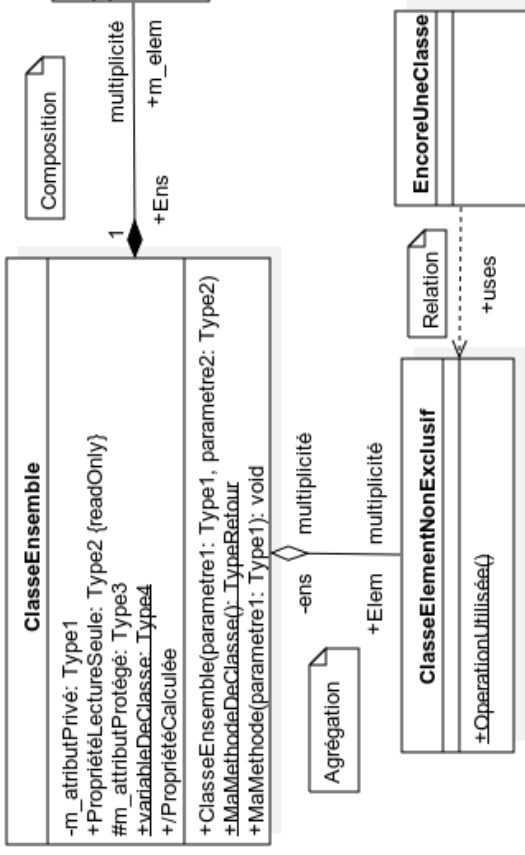
- * relation entre une entreprise et un de ses services
- * Une entreprise peut être vue comme un ensemble de services
- * Si l'entreprise disparaît, le service n'existe plus et il disparaît aussi.
- * Si l'entreprise est réorganisée, le service peut disparaître, même si l'entreprise survit

La relation entre ClasseEnsemble et ClasseElementNonExclusif est une agrégation

- Un élément peut être associé à plusieurs ensemble
- Un ensemble peut contenir plusieurs éléments
- La multiplicité peut être quelconque des deux côtés suivant les applications

Exemple de relation d'agrégation :

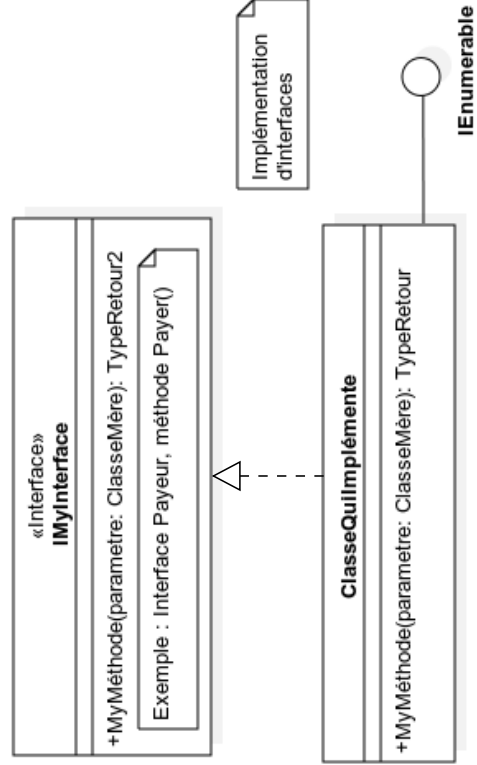
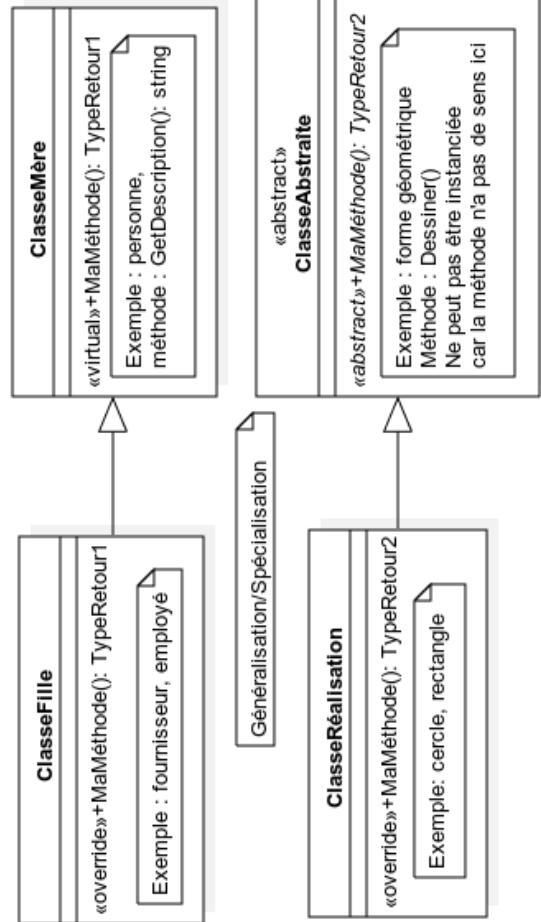
- * Relation entre une société de services et un client
- * La société de service peut être vue comme un ensemble de client
- * Un client peut faire appel à plusieurs sociétés de services
- * Une société de services peut avoir plusieurs clients



La multiplicité d'un attribut, obligatoire, apparaît du côté du type de l'attribut en question.
 Par exemple, la multiplicité du composite Ens est 1

Exemples de multiplicités :

- 0..* (nombre quelconque supérieur ou égal à 0, une entreprise peut avoir 0 clients)
- 1..* (ensemble obligatoirement non vide, un auteur doit avoir 1 livre)
- 1 (un seul objet, pas de tableau ni collection)
- 0..1 (zéro ou un objet, type nullable par exemple)
- 0..10 (entre 2 et 10 objets, par exemple, voitures garées dans un parking à 10 places)

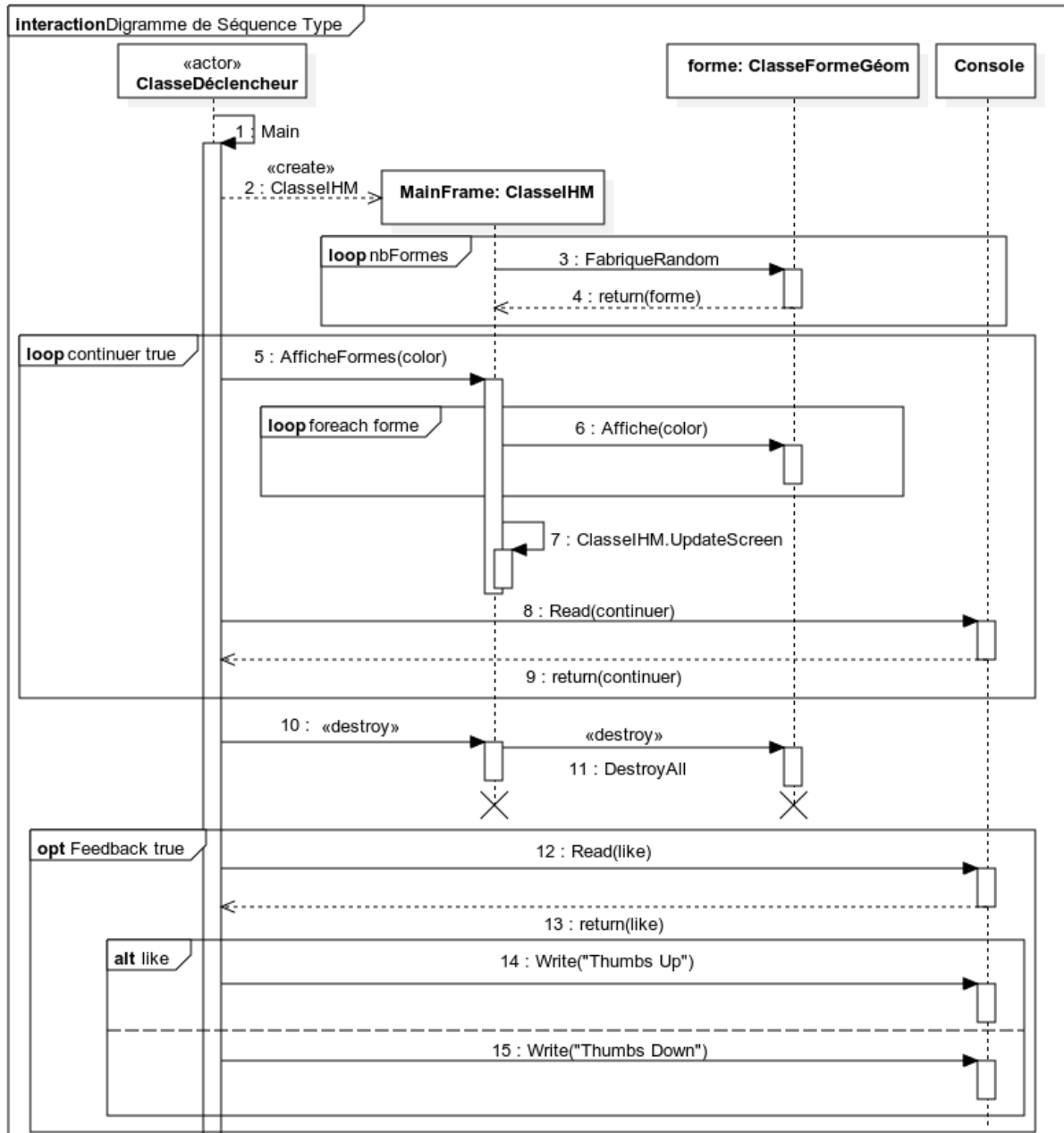
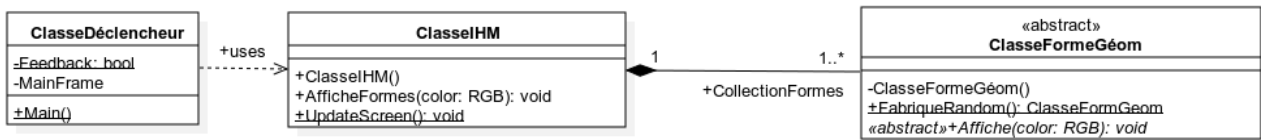


Conventions UML

Diagrammes de Séquence

Rémy Malgouyres, Université d'Auvergne

Voici un petit Diagramme de Classes et un Diagramme de Séquence pour le déroulement d'une animation. Le programme affiche des formes géométriques tant que l'utilisateur souhaite continuer. À la fin, si l'option *feedback* est active, l'utilisateur peut donner son opinion (*liker*).



```

ClasseDéclencheur.Main(){
    while (continuer){
        MainFrame.AfficheFormes(color);
        Console.Read(continuer);
    }
    if (Feedback){
        Console.Read(like);
        if (like)
            Console.Write("Thumbs Up");
        else
            Console.Write("Thumbs Down");
    }
}

```

```

class ClasseIHM{
    ...
    public void AfficheFormes(color:RGB){
        foreach (ClasseFormeGeom forme
            in CollectionFormes){
            forme.Affiche(color);
        }
        UpdateScreen();
    }
    ...
}

```