

## Cartouche du document

**Année : ING 1**  
**Matière : MethoI**  
**Activité : Travail dirigé**

## Objectifs

L'objectif de ce T.D. est d'approfondir les notions d'objets, de classes et de diagrammes de classes dans le but de modéliser l'aspect statique et structurel d'un projet donné.

## Sommaire des exercices

- 1 - Encapsulation des objets
- 2 - Une modélisation à connaître
- 3 - Une modélisation corsée mais à connaître
- 4 - Modélisation d'un circuit logique

## Corps des exercices

### 1 - Encapsulation des objets

#### Énoncé :

On considère la classe PointEcran. Tous les objets de cette classe représentent des points sur un écran. Les caractéristiques de l'écran sont connues au début de l'exécution du programme qui gèrent ces objets.

#### Question 1)

##### Énoncé de la question

Trouver les attributs cette classe. Cette classe devra gérer la taille (commune à tous les objets) de l'écran.

##### Solution de la question

```
x : Entier  
y : Entier  
xMax : Entier  
yMax : Entier
```

#### Question 2)

##### Énoncé de la question

Donner les caractéristiques des différents attributs pour que l'objet soit toujours dans un état

cohérent.

### Solution de la question

`0 <= x <= xMax.`  
`0 <= y <= yMax.`  
`x et y sont des attributs d'objet.`  
`xMax et yMax sont des attributs de classes.`  
`xMax et yMax sont des attributs gelés.`

### Question 3)

#### Énoncé de la question

On s'intéresse aux méthodes de la classe `PointEcran` qui change l'état de l'objet.

Que doit faire systématiquement le programmeur dans le code de la méthode pour que l'objet reste dans un état cohérent.

#### Solution de la question

Après les calculs de `x1` et/ou `y1` qui seraient les nouvelles valeurs `x` et/ou `y`, il doit vérifier que les résultats de ces calculs sont dans les limites autorisées, c'est-à-dire que `0 <= x1 <= xMax` et `0 <= y1 <= yMax`.

Si cela n'est pas le cas, il doit avoir une règle de gestion (voir cahier des charges) pour rectifier les valeurs de `x1` et `y1`.

Typiquement, le programmeur pourrait écrire une méthode privée qui fait la rectification éventuelle d'un mauvais couple. Cette méthode serait systématiquement appelée à la fin de chaque opération qui change l'état du point.

### Question 4)

#### Énoncé de la question

Pourquoi les attributs qui positionnent le point dans l'écran doivent être privés ?

#### Solution de la question

En étant privés, on s'assure que pour changer l'état du point, il faudra appeler une des méthodes de classe `PointEcran` qui le permet. Hors dans les réponses précédentes, on a expliqué comment on devait coder ces méthodes pour qu'une instance de la classe `PointEcran` reste dans un état cohérent.

## 2 - Une modélisation à connaître

### Question 1)

#### Énoncé de la question

On veut décrire une personne dans l'univers à travers son nom, son prénom et son âge. De plus, on désire connaître son père et sa mère biologique.

Donner le diagramme de classes de ce problème

#### Solution de la question

Une classe pour Personne : nom, prénom, âge.  
Deux classes pour Homme et Femme qui dérive de Personne.

Une association nommée estPereDe entre Personne et Homme. Le rôle de l'instance Personne est enfantDe.  
Le rôle de l'instance Homme est pereDe.

Une association nommée estFilsDe entre Personne et Femme. Le rôle de l'instance Personne est enfantDe.  
Le rôle de l'instance Femme est mereDe.

## 3 - Une modélisation corsée mais à connaître

### Énoncé :

On considère un logiciel qui permet de créer des composants graphiques. Un composant graphique peut être :

- un bouton;
- un champ d'édition;
- un label.
- une fenêtre.

Une fenêtre est un ensemble de composants graphiques.

### Question 1)

#### Énoncé de la question

Trouver une modélisation de ce problème.

#### Solution de la question

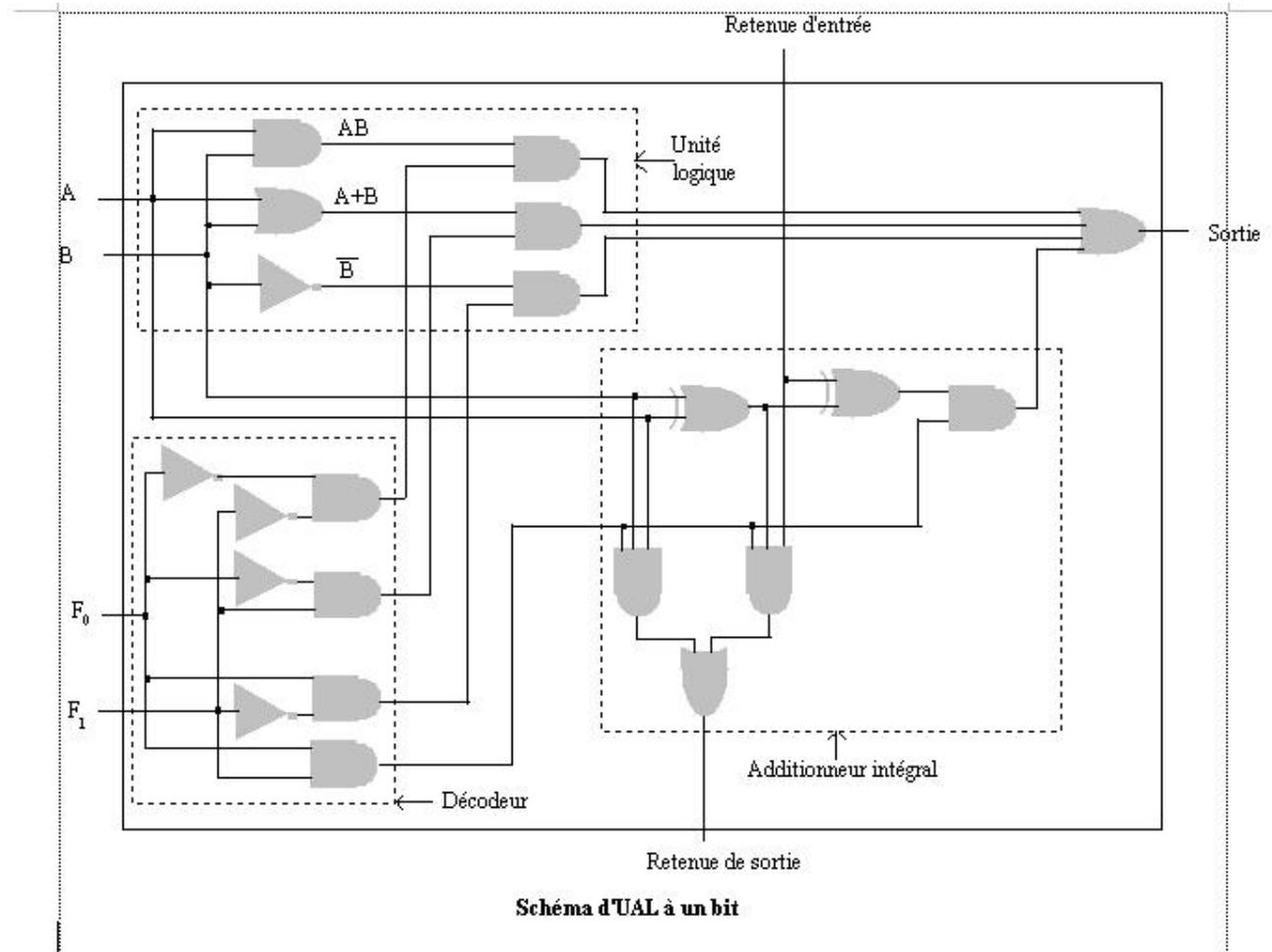
Une classe Abstraite ComposantGraphique  
Une classe Triangle qui hérite de ComposantGraphique.  
Une classe Cercle qui hérite de ComposantGraphique.  
Une classe Polygone qui hérite de ComposantGraphique.  
Une classe Fenetre qui hérite de ComposantGraphique et qui est une composition

de ComposantGraphique.

## 4 - Modélisation d'un circuit logique

### Énoncé :

Nous nous intéressons à la description du comportement d'un circuit logique donné. Nous traitons 3 sortes de portes logiques : Le NOT, le AND et le OR.



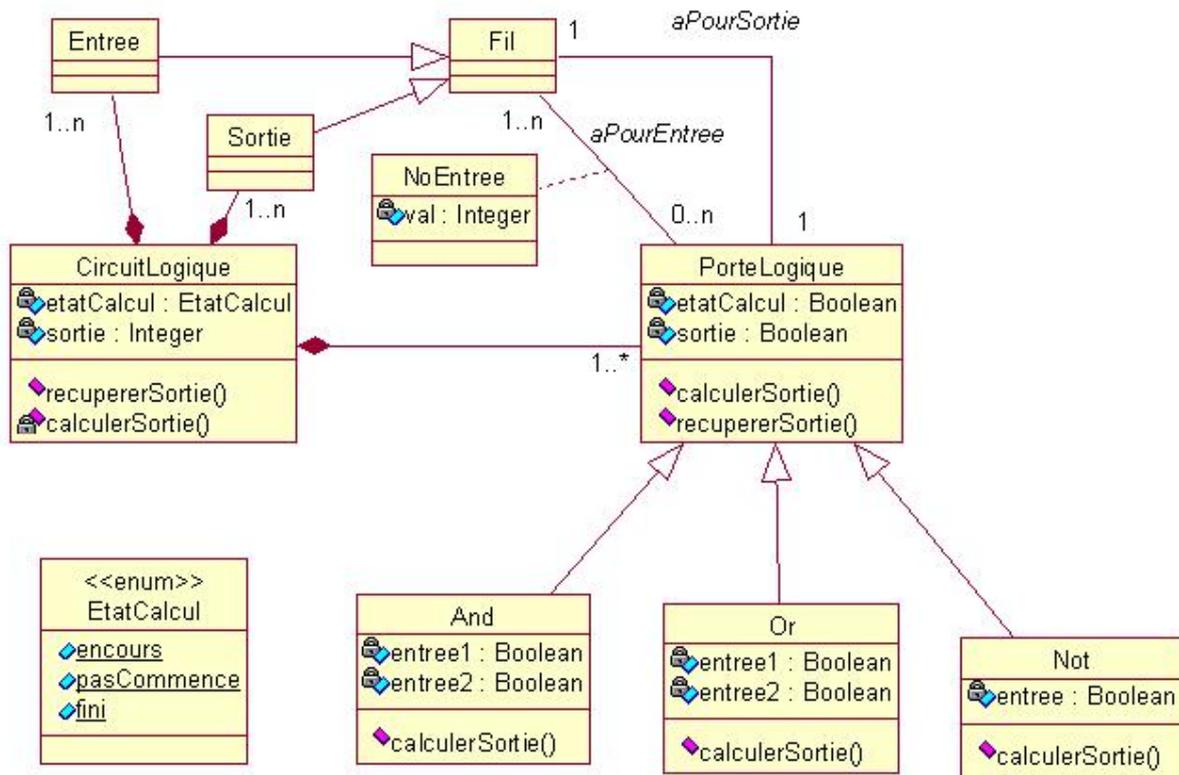
## UAL à un bit

### Question 1)

Énoncé de la question

Compléter le diagramme de classes des circuits logiques.

Solution de la question



**Diagramme de classes d'un circuit logique**