

**Université Paris XII  
IUT de Fontainebleau  
Département informatique  
FI-2  
2000-2001**

## **PARTIEL 1**

Seuls sont autorisés, à titre de documents, les listings comprenant explicitement le nom de l'étudiant (et celui-ci seulement) et les notes manuscrites.

### Exercice 1.- (Arithmétique modulaire)

- 1°) Définir une classe `ModClass` des entiers congrus modulo 15 (soit  $\mathbb{Z}/15\mathbb{Z}$ ). Le seul membre donnée `val` est un entier de 0 à 14. Les méthodes sont le constructeur par défaut, le constructeur admettant un argument entier (`val` est alors le reste modulo 15 de cet entier), une méthode d'affichage et les opérateurs surchargés `+`, `-` et `*`.

- 2°) Écrire un programme C++ permettant de tester cette classe.

### Exercice 2.- (Classe générique)

- 1°) Définir une classe générique `Stock` de variable de type `T` dont les donnée sont un tableau de cinq élément de type `T` et un `index` de type `int` et dont les méthode sont : le constructeur (par défaut) initialisant l'index à 0 ; la méthode `insert` prenant un argument de type `T` et plaçant cet élément à la position indiquée par l'index si celui-ci est inférieur à 4 (elle renvoie aussi le booléen vrai dans ce cas) et renvoyant le booléen faux sinon ; la méthode `trouve` prenant un argument de type `T` et renvoyant l'index de cet élément s'il se trouve dans le tableau et -1 sinon ; la méthode `get` prenant comme argument un entier et renvoyant l'élément qui se trouve à cet index ou un message d'erreur si l'entier n'est pas dans l'intervalle voulu.

- 2°) Tester cette classe générique avec des instantiations entière, réelle et caractère.

### Exercice 3.- (Héritage)

Attention! Commencer par lire l'énoncé en entier pour éviter à avoir à revenir sur les définitions des différentes classes après coup.

- 1°) Concevoir une classe **figure** n'ayant aucune donnée mais une méthode **aire** sans argument et renvoyant un réel (nul).

- 2°) Concevoir une classe dérivée **ellipse** dont les données sont deux réels, le petit axe  $a$  et le grand axe  $b$ , un constructeur par défaut, un constructeur ayant deux arguments (les valeurs du petit axe et du grand axe), la méthode **aire** étant surchargée (rappelons que l'aire d'une ellipse est  $S = \pi.a.b$ ).

- 3°) Concevoir une classe dérivée **cercle** de la classe **ellipse**, possédant un constructeur à un argument réel (le rayon) et la méthode **perimetre**.

- 4°) Concevoir une fonction **affiche\_aire** qui prend en argument un pointeur sur un objet de la classe **figure** (y compris de ses classes dérivées) et qui affiche l'aire de l'objet.

- 5°) Écrire une programme C++ permettant de tester ces classes et cette fonction.