

## Consignes relatives au déroulement de l'épreuve

Date :	25 avril 2014
Contrôle de :	<b>4IRC - Module « Conception orientée Objet – DP 1 » - 1<sup>ère</sup> session</b> <b>1<sup>ère</sup> partie</b>
Durée :	25 minutes
Professeur responsable :	Françoise Perrin
Documents :	<b>documents et calculatrices interdits</b>

LES TELEPHONES PORTABLES ET AUTRES APPAREILS DE STOCKAGE DE DONNEES NUMERIQUES NE SONT PAS AUTORISES.

Les téléphones portables doivent être éteints pendant toute la durée de l'épreuve et rangés dans les cartables.

Les cartables doivent être fermés et posés au sol.

Les oreilles des candidats doivent être dégagées.

## Rappels importants sur la discipline lors des examens

La présence à tous les examens est strictement obligatoire ; tout élève présent à une épreuve doit rendre une copie, même blanche, portant son nom, son prénom et la nature de l'épreuve.

Une absence non justifiée à un examen invalide automatiquement le module concerné.

Toute suspicion sur la régularité et le caractère équitable d'une épreuve est signalée à la direction des études qui pourra décider l'annulation de l'épreuve; tous les élèves concernés par l'épreuve sont alors convoqués à une épreuve de remplacement à une date fixée par le responsable d'année.

Toute fraude ou tentative de fraude est portée à la connaissance de la direction des études qui pourra réunir le Conseil de Discipline. Les sanctions prises peuvent aller jusqu'à l'exclusion définitive du (des) élève(s) mis en cause.

## 1<sup>er</sup> exercice : 3 points

Répondre aux questions suivantes de manière précise mais synthétique (3 lignes maximum)

1. Que sont les Design Patterns et à quoi servent-ils ?
2. Que signifie le principe « Programmez des interfaces et non des implémentations » ?
3. A quoi sert une fabrique ?

## 2<sup>ème</sup> exercice : 3 points

Soit une classe qui a des données et un comportement spécifique et qui contient un objet instance de la même hiérarchie de classe ou pas. Exemple :

```
public class ClasseQuiContient1Objet /* extends X implements Y */{  
    private T ObjetContenu; // T peut être X ou Y ou tout autre classe/interface  
  
    // ... autres attributs et méthodes spécifiques de ClasseQuiContient1Objet  
}
```

Cette architecture modélise la mise en œuvre d'un Design Pattern decorator, proxy, adapter, strategy, etc.

**Dans lesquels de ces contextes (decorator, etc.) est il pertinent de définir, ou pas, un accesseur (getObjetContenu) et un mutateur (setObjetContenu) sur ObjetContenu ? Justifiez.**