Université de Jijel
Faculté des sciences exactes et informatique
Département d'informatique

Correction de l'examen de POO

Question de cours

A. Répondez aux questions suivantes (6 points)

1. **Quelle est la différence entre un objet et une classe ?**

- Une classe est une description d'un ensemble d'objets. Un objet est une instanciation d'une classe (0.5)
- Une classe correspond à la généralisation de la notion de type que l'on rencontre dans les langages classiques. Les objets apparaissent comme des variables d'un tel type classe (0.5)
- Une classe est un modèle d'objet. Un objet a une réalité matérielle car il possède des champs avec valeurs. (0.5)

2. **Quels sont les avantages de l'abstraction ?**

Les avantages de l'abstraction

- Réduire la complexité. (0.5)
- Évite la duplication de code et augmente la possibilité de réutilisation. (0.5)
- Aide à renforcer la sécurité d'une application ou d'un programme car seuls les détails importants sont fournis à l'utilisateur. (0.5)

3. **Peut-on instancier une interface ?** Non. (0.5)
4. **Peut-on instancier une classe abstraite ?** Oui. (0.5)
5. **Par quoi les objets interagissent ?**

   Les objets interagissent par envoie de messages (1point), qui est un appel de fonction. (1 point)

B. Répondez par juste / faux et corrigez la faute (10 points)

Question	Réponse
1. Tous les champs d'une interface sont public **static final**.	**1. Juste** (0.5)

2.
interface MonInterface {
public int x;
public void static f();
}
	2. faux (0.5)
	Les seules variables que l'on peut définir dans une interface sont des variables de classe qui doivent être constantes. (0.5)
	interface MonInterface {
	public int **x=2;** (0.5)
	public void static f();
	}

3. Une classe ne peut pas implanter plusieurs interfaces, et ne peut avoir qu'une seule superclasse.	**3. faux** (0.5) Plusieurs interfaces peuvent être implémentées dans une même classe. Mais une classe ne peut avoir qu'une seule superclasse (1)
4. Déclaration d'une constante avec le modificateur **final**	**5. Faux** (0.5) La déclaration d'une constante : **public static final** (1)
6. Appeler le constructeur d'une classe mère **this**.	6. **faux,** (0.5) par **super** (1)
7. On peut redéfinir une méthode (polymorphisme de redéfinition) qualifié **private.**	7. **Juste** (0.5)
8. Dans la programmation orienté objet, on cherche à répondre à la question "Que doit faire mon programme ?	8. **faux** (0.5) Dans la programmation orienté objet, on Répond à la question : de quoi doit être composé mon programme ? (1)
9. Un objet incorpore des aspects statiques présentés par les procédures et dynamiques par les données.	9. **Faux** (0.5) Un objet incorpore des aspects statiques présentée par **les données** et dynamiques présentée par **les procédures** au sein d'une même notion d'objet. (1)

Exercice : Soit le code de la classe **rectangle** **(4 points)**

1. Corriger les fautes existantes.
2. Compléter les fonctions de la classe :
   - Constructeur d'initialisation, qui permet d'initialiser les attributs de la classe.
   - Les fonctions **GetLargeur** et **GetLongueur**.
   - La fonction **surface** pour avoir la surface d'un rectangle.
3. **Carre** est une classe qui hérite de la classe **Rectangle**. Donner le code pour exprimer cet héritage

Code

```
public class rectangle {
private int Largeur ;
private int Longueur ;

public rectangle (int x, int y) {


}

public int GetLargeur() {

}

public int GetLongueur() {
}

public int surface() {

}}
```

Correction

```
public class Rectangle extends Forme {          (0.25)
private int largeur ;                           (0.25)
private int longueur ;                          (0.25)

public Rectangle(int x, int y) {                (0.25)

this.largeur = x ;                              (0.25)
this.longueur = y ;                             (0.25)
}

public int getLargeur() {                       (0.25)
return this.largeur ;                           (0.25)
}

public int getLongueur() {                      (0.25)
return this.longueur ; (                        0.25)
}

public int surface() {                          (0.25)
return this.longueur * this.largeur ;
}}      (0.25)          (0.25)
```

(0.25) sur les points virgules et organisation.

```
3. class Carre extends Rectangle                (0.5)

   {....}
```