Examen POO

Groupe : *RSI21* *Date : 04/01/2023* *Durée : 1h30*

Enseignant : **M. Hedi HMANI** *Documents non autorisés*

Barème approximatif : *2 + 4 + 14 points*

Exercice 1:

Cocher la bonne réponse dans chacune des questions suivantes :

a.

- ☑ Une classe peut implémenter plusieurs interfaces mais doit étendre une seule classe
- ☐ Une classe peut implémenter plusieurs classes mais doit étendre une seule interface
- ☐ Une classe peut implémenter plusieurs classes et peut étendre plusieurs interfaces
- ☐ Une classe doit implémenter une seule interface et étendre une seule classe

b. Pour empêcher une classe d'être étendue, nous utilisons le mot clé :

- ☐ *abstract*
- ☐ *static*
- ☑ *final*
- ☐ *protected*

Exercice 2 :

Quel sera le résultat de l'exécution de chacun des programmes suivants. Cocher la bonne réponse.

Programme 1 :

```
class Animal {
     String getType() {
          return "animal";
     }
}// fin classe Animal

class Felin extends Animal {
     String getType() {
          return "felin";
     }
}// fin classe Felin

class Chat extends Felin {
     String getType() {
          return "chat";
     }
}// fin classe Chat
```

```java
public class Test {
	public static void main(String[] args) {
		Chat ch1 = new Animal();
		Chat ch2 = new Felin();
		System.out.println(ch1.getType() + " " + ch2.getType());
	}
}
```

- ☐ animal felin
- ☐ chat chat
- ☑ Erreur de compilation
- ☐ Erreur à l'exécution

Pogramme 2 :

```java
class MaClasse {
	int i;
}

public class Test {
	public static void main(String[] args) {
		MaClasse a = new MaClasse();
		MaClasse b = new MaClasse();
		a.i = 10;
		b = a;
		b.i = 5;
		System.out.println(a.i);
	}
}
```

- ☑ 5
- ☐ 10
- ☐ 15

Programme 3 :

```java
class Arbre {
	public Arbre() {
		System.out.println("Arbre ");
	}
}

class Pommier extends Arbre {
	public Pommier() {
		System.out.println("Pommier ");
	}
}
```

```java
public class Test {
     public static void main(String[] args) {
          Arbre a = new Pommier();
     }
}
```

- ☐ Arbre
- ☐ Pommier
- ☑ Arbre Pommier
- ☐ Erreur de compilation
- ☐ Erreur à l'exécution

Programme 4 :

```java
class Individu {String nom ; }

public class Test {
     public static void main(String[] args) {
          Individu tab[] ;
          Individu ind = new Individu();
          ind.nom = "Ben Salah" ;
          tab[0] = ind;
          System.out.println (tab[0].nom);
     }
}
```

- ☐ Ben Salah
- ☐ Exception de type ArrayIndexOutOfBoundsException
- ☐ Exception de type NullPointerException
- ☑ Erreur de compilation
- ☐ Erreur à l'exécution

Correction du problème :

Element.java

```java
package gestionPubs;

public abstract class Element {//1
     private String code;

     public Element(String code) {
          this.code = code;
     }
}
```

Texte.java

```java
package gestionPubs;

public class Texte extends Element{
    private String contenu;

    public Texte(String code, String contenu) {
        super(code);
        this.contenu = contenu;
    }
}
```

Image.java

```java
package gestionPubs;

public class Image extends Element{
    private String url;

    public Image(String code, String url) {
        super(code);
        this.url = url;
    }
}
```

Video.java

```java
package gestionPubs;

public class Video extends Element {//2
     private String description;
     private String url;

     public Video(String code, String desc, String url) {
          super(code);
          this.description = desc;
          this.url = url;
     }

     @Override
     public String toString() {
          return "[Video : " + description + " ; " + url + "]";
     }
}
```

Utilisateur.java

```java
package utils;

public class Utilisateur {// 7
	public static final int NB_MAX_AMIS = 5000;
	private String nom;
	private String prenom;
	private String email;
	private Utilisateur[] amis = new Utilisateur[NB_MAX_AMIS];
	private int nbreAmis = 0;

	public Utilisateur(String nom, String prenom, String email) {
		this.nom = nom;
		this.prenom = prenom;
		this.email = email;
	}

	@Override
	public boolean equals(Object obj) {
		if (this == obj)
			return true;

		if (obj == null)
			return false;
		if (getClass() != obj.getClass())
			return false;
		Utilisateur other = (Utilisateur) obj;
		if (email == null) {
			if (other.email != null)
				return false;
		} else if (!email.equals(other.email))
			return false;
		return true;
	}

	public boolean chercher(Utilisateur tab[]) {
		if (tab == null)
			return false;
		for (Utilisateur ut : tab)
			if (ut != null && ut.equals(this))
				return true;
		return false;
	}

	public boolean ajouterAmi(Utilisateur u) {
		if (u==null||u==this||nbreAmis==NB_MAX_AMIS||u.nbreAmis==NB_MAX_AMIS)
			return false;
```

```
        if (u.chercher(amis))
            return false;
        for (int i = 0; i < amis.length; i++)
            if (amis[i] == null) {
                amis[i] = u;
                this.nbreAmis++;
                u.ajouterAmi(this);
                break;
            }
        return true;
    }

    public boolean retirerAmi(Utilisateur u) {
       if(u!=null)
          if (u.chercher(amis)) {   //ou chercher(u.amis)
              //suppression de u des amis de this
              int i = 0;
              boolean trouve = false;
              while (i < nbreAmis && !trouve) {
                  if (amis[i].equals(u))
                      trouve = true;
                  else
                      i++;
              }
              if (trouve) {
                  for (int j = i; j < amis.length - 1; j++) {
                      amis[j] = amis[j + 1];
                  }
                  nbreAmis--;
              }

              //suppression de this des amis de u
              u.retirerAmi(this);
              return true;
          }
       return false;
    }
}
```

Mur.java

```
package utils;

public class Mur {//3
     private Utilisateur proprietaire;
     private boolean estActif = true;
```

```
    public Mur(Utilisateur proprietaire) {
        this.proprietaire = proprietaire;
    }

    public void activer() {
        this.estActif = true;
    }

    public void desactiver() {
        this.estActif = false;
    }

    public boolean isActif() {
        return estActif;
    }

    public Utilisateur getProprietaire() {
        return proprietaire;
    }
}
```

Test.java

```
package utils;

import gestionPubs.Element;
import gestionPubs.Image;
import gestionPubs.Texte;
import gestionPubs.Video;

public class Test { //2
    public static void main(String[] args) {

        //a.   créer deux utilisateurs user1 et user2,
        Utilisateur user1 = new Utilisateur("Mohamed", "Ben Saleh",
"mbs@gmail.com");
        Utilisateur user2 = new Utilisateur("Ali", "Ben Salem",
"abs@gmail.com");
        //b.   ajouter une relation d'amitié entre user1 et user2.
        user1.ajouterAmi(user2);   // ou user1.ajouterAmi(user2)
        //c.   créer un mur m1 pour user1 et un mur m2 pour user2,
        Mur m1 = new Mur(user1);
        Mur m2 = new Mur(user2);
        //d.   créer un tableau d'éléments et alimenter le avec des
instances de votre choix.
        Element[] tab = new Element[5];
        tab[0] = new Texte("T01", "Première publication");
```

```
        tab[1] = new Image("I01", "https://isetsf.rnu.tn/isetsf.png");
        tab[2] = new Image("I02", "https://isetsf.rnu.tn/techinfo.png");
        tab[3] = new Video("V01", "Iset Sfax",
"https://isetsf.rnu.tn/isetsf.png");
        tab[4] = new Video("V02", "Dep Tech Info",
"https://isetsf.rnu.tn/techinfo.png");
        //e.   afficher séparément les éléments selon leurs types : les
vidéos, les images et les textes.
        System.out.println("Liste des textes");
        for (int i = 0; i < tab.length; i++) {
            if (tab[i] instanceof Texte)
                System.out.println(tab[i]);
        }
        System.out.println("Liste des images");
        for (int i = 0; i < tab.length; i++) {
            if (tab[i] instanceof Image)
                System.out.println(tab[i]);
        }
        System.out.println("Liste des videos");
        for (int i = 0; i < tab.length; i++) {
            if (tab[i] instanceof Video)
                System.out.println(tab[i]);
        }
    }
}
```