UEF4.3. Programmation Orientée Objet
Contrôle Intermédiaire
1h45 - Documents & Téléphone interdits

Exercice 1 (4,5 pts)

Indiquez pour chacun des codes suivants:

- S'il compile ou pas.
- Si la réponse est oui, indiquez ce qu'il affiche à son exécution,
- Si la réponse est non :
  - expliquez pourquoi,
  - proposez, quand c'est possible, des corrections puis indiquez ce qu'il affiche à son exécution.

i./

```
class A{
int i;
A(int i){ this.i = i; }
}
class B extends A{
A a;
int i = 2;
B(){ this.a = this; }
B(A a){
super(3);
this.a = a;
}
public static void main(String[]
args){
A a2 = new A(5);
B b1 = new B(a2);
B b2 = new B();
System.out.println(b1.i);
System.out.println(((A) b1).i);
System.out.println(b1.a.i);
System.out.println(b2.i);
System.out.println(((A) b2).i);
System.out.println(b2.a.i);
}
}
```

ii./

```
public class Test
public static void main ( String
args[ ])
int k ; int j=0;
try
k = 1 / j;
throw new
  IndexOutOfBoundsException();
catch(RuntimeException ex )
System.out.println(" Runtime
  Exception");
catch(IndexOutOfBoundsException
ex)
System.out.println("Dépassemnt
de la capacité" );
catch(ArithmeticException ex)
System.out.println("Exception
  arithmétique " );
finally {
System.out.println(" fin");
 }
System.out.println(" suite");
```

iii./

```
package z;
public class Test {
public static void main ( String args[ ]) {
    D d=new D(3);
    System.out.println(d.getI());
}
}
```

```
package p;
interface I {
public int getI();
}
```

```
package q;
abstract class C
implements I {
private int i;
C(int i) {
this.i = i;
}
}
```

```
package q;
public class D
extends C {
D(int x) {
super(x);
this.i = 0;
}
}
```

Exercice 2 (8 pts)

Nous souhaitons gérer une hiérarchie de moyens de transport, où les classes *Moto*, *Voiture*, *Bus* dérivent de la classe *MoyenTransportRoutier* et les classes *Tramway*, *Metro*, *Train* dérivent de la classe *MoyenTransportFerrovier*. Les classes *MoyenTransportRoutier* et *MoyenTransportFerrovier* dérivent de la classe *MoyenTransport*.

1. Quelles sont les affections correctes parmi ces quatre propositions ?
   - **a.** MoyenTransportRoutier [] tab = new Bus [10] ;
   - **b.** Voiture [] tab = new Tramway [10] ;
   - **c.** MoyenTransport[] tab = new MoyenTransport [10] ;
   - **d.** MoyenTransportRoutier [] tab = new MoyenTransport [10] ;

2. Laquelle des propositions précédentes permet de créer un tableau pouvant contenir des objets de type Voiture et Tramway (a, b, c ou d)?

3. Nous supposons que la méthode afficher() est définie dans la classe *MoyenTransport* et redéfinie dans les classes *MoyenTransportRoutier*, *Tramway* et *Train* . Dites quelles méthodes *afficher()* seront appelées suite à l'exécution du code suivant :

```
public class Test {
public static void main(String args[]) {
 MoyenTransport m1 = new MoyenTransport(); MoyenTransport m2 = new Train();
 MoyenTransportRoutier m3 = new Moto();
 MoyenTransportFerrovier m4 = new Tramway();  Train m5 = new Train();
 MoyenTransportFerrovier m6 = new Metro();    Moto m7=new Moto();
 m1.afficher(); m2.afficher(); m3.afficher();
 m4.afficher(); m5.afficher(); m6.afficher(); m7.afficher();
 ((MoyenTransport)m3).afficher();((MoyenTransportFerrovier)m5).afficher();
 }
}
```

4. En considérant les initialisations faites dans le précédent code, indiquez pour chacune des instructions suivantes :

- si elle est correcte ou si elle provoque une erreur à la compilation ou à l'exécution.

- si on peut résoudre le problème de la compilation par un cast (et donnez ce cast), et si oui, s'il reste une erreur à l'exécution ou non.

```
1. m1 = m2;          2. m4 = m2;          3. m3 = m1;          4. m4 = m5;
5. m5 = (Train)m2;   6. m6 = m7;          7. m7=m3;            8. m6=(Metro)m2;
```

Exercice 3 (7,5 points)

On veut écrire un programme gérant le salaire des enseignants d'une université. Chaque enseignant a un nom, un prénom et un certain nombre d'heures de cours assurées dans l'année. Il existe plusieurs catégories d'enseignants. Les enseignants-chercheurs sont payés avec un salaire fixe plus le revenu des heures complémentaires (un revenu fixe pour chaque heure effectuée). Les heures complémentaires sont les heures effectuées au-delà de 192h. Ainsi, un enseignant-chercheur qui donne 200h de cours dans l'année recevra (salaireFixe*12+ revenuHeureComplémentaire*8) DA sur l'année. Les vacataires sont des enseignants venant d'un organisme extérieur à l'université. Cet organisme est décrit par une chaîne de caractères. Les vacataires sont payés de l'heure selon un revenu fixe spécifique (revenuHeureVacataire). Les étudiants en post-graduation (doctorants) peuvent assurer des cours et sont payés selon un revenu fixe propre à leur statut (revenuHeureDoctorant), mais ne peuvent dépasser un certain nombre d'heures de cours (nombreHeuresMax). Un doctorant qui fait plus d'heures que le nombre maximum ne recevra que le salaire correspondant au nombre maximum de cours. Un début de code a été écrit par un programmeur débutant en POO.

```
public class Enseignant {

 public String nom;
 public String prenom;
 public int nbHeures;
 int type; //0: enseignant-Chercheur, 1: enseignant vacataire, 2: doctorant
 public final double salaireFixe ;
 public final double revenuHeureComplementaire;
 public final double revenuHeureVacataire;
 public final double revenuHeureDoctorant;
 public final int nbHeuresMax ;

// Constructeur
public Enseignant (String nom, String prenom, int nbHeures, int type) {
   this.nom = nom;
   this.prenom = prenom;
   this.nbHeures = nbHeures;
   this.type = type;
 }

public double calculerSalaire() {
  if (type == 0 ){...} //calculer le salaire d'un enseignant-chercheur;
  else if (type == 1) {...}//calculer le salaire d'un enseignant vacataire
  else if (type == 2) {...} //calculer le salaire d'un doctorant
  else return 0; // si le type est différent de 0, 1 et 2
 }
}
```

1. Quels sont les principes de la POO non respectés dans ce code ? expliquez (Toute réponse sans explication sera rejetée)

2. Proposez un nouveau programme conforme aux principes de la POO qui calcule le salaire de chaque enseignant.
3. Expliquez les avantages du programme que vous avez proposé dans la question précédente.
4. Donnez l'instruction qui permet de déclarer un tableau contenant à la fois des références vers des enseignants-chercheurs, des enseignants vacataires et des doctorants. Quel principe OO avez-vous utilisé ?
5. Donnez l'instruction permettant de calculer les salaires de tous les éléments du tableau précédent. Quelle notion OO est utilisée dans cette instruction ?
6. On souhaite que le programme affiche un message lorsqu'un doctorant dépasse le nombre d'heures maximum. Expliquez (sans donner le code source) par quel moyen cela pourrait être implémenté.