(ISIL A + ISIL B + ACAD C)

Exercice 1 (5 pts)

On définit les classes et interfaces suivantes :

```
public interface I1 { void f();}
public interface I2 {void g();}

public abstract class A implements I1,I2{
protected  int x=0;
protected static int y=0,  z=0;

public A(int x) {this.x=x ; }
public A(){}
public void h(){x++;};

public void f(){z=x; z=z*3;}
public abstract void g() ;
}
class B extends A{      public void g(){y=x; y++;}  }

class C extends B{      public void f(){ z++;}        public void g(){ y=y*4;}  }

public class Application {

      public static void main(String[] args) {
       A[] tab=new A[2];          tab[0]=new B();          tab[1]=new C();

       for (A e:tab){     e.h();     e.f();   e.g();}

       System.out.println("y = "+ A.y+   " z = "+A.z);
      } }
```

1/ Sélectionner les réponses correctes et corriger les réponses fausses :

a. Une classe abstraite ne peut pas avoir de constructeurs puisqu'elle ne peut pas être instanciée.
b. Un constructeur avec paramètres doit être rajouté à la classe B, un autre à la classe C.
c. Un objet de la classe B a uniquement les types : B, A et Object.
d. La méthode h () est polymorphe : on peut lui appliquer le polymorphisme.
e. La méthode f () est polymorphe : on peut lui appliquer le polymorphisme.
f. Les types statiques des objets tab[0] et tab[1] sont B et C.
g. Le type dynamique de tab[0] est A.
h. C'est le concept de l'héritage qui nous permet de sélectionner dynamiquement la méthode à exécuter.

2/ Dérouler la méthode main

Exercice 2 (15 pts)

Définir une classe **Monôme** composée de deux attributs *coefficient* de type réel et *degré* de type entier.
On rappelle qu'un monôme s'écrit sous la forme $a\,x^n$ tel que a représente le coefficient et n le degré mais le x n'est pas représenté dans la classe.

Compléter la classe par les méthodes suivantes :

- Un constructeur avec deux paramètres.

- Un constructeur avec un seul paramètre réel qui permet de créer un monôme avec un degré nul.

- Une méthode *dérivé ()* qui permet de retourner le monôme dérivé
- Une méthode *évaluer (float x)* qui retourne la valeur du monôme pour la valeur x ;
- Redéfinir la méthode *toString* de telle sorte qu'elle retourne le monôme sous cette forme : *ax^b* exemple 5x^3
- Une méthode afficher()
- Implémenter l'interface Comparable tel que :
  - Deux monômes sont égaux s'ils ont les mêmes coefficients et les mêmes degrés
  - Un monôme est plus grand qu'un autre s'il a un degré supérieur ou bien le même degré et un coefficient supérieur

1. Définir une classe **Polynôme** composé d'un attribut *poly* de type liste d'objets Monôme (*ArrayList*). Compléter la classe par les méthodes suivantes :
   - Un constructeur avec un seul paramètre, qui permet d'initialiser *poly* et de le trier en utilisant la méthode statique *sort* de la classe prédéfinie *Collections.*
   - Une méthode *degré* qui retourne le degré du polynôme correspondant au plus grand degré de ses monômes.
   - Une méthode *dérivé* qui retourne le polynôme dérivé.
   - Une méthode *évaluer (float x)* qui retourne la valeur du polynôme pour la valeur *x* .
   - Redéfinir la méthode *toString* de telle sorte qu'elle retourne le polynôme sous la forme d'une somme de monômes: exemple, 5x^3+ 2x^7
   - Implémenter l'interface *Comparable* tel que :
     - Deux polynômes sont égaux s'ils sont composés de monômes égaux.
     - Un polynôme est plus grand qu'un autre : (1) s'il a un degré supérieur ou bien (2) le même degré et le coefficient correspondant est supérieur ou bien (3) les deux polynômes ont une partie des degrés supérieurs qui est commune et la partie restante des monômes du premier polynôme est supérieure.

       Exemple : soit $p1 = 2x^7 + 5x^2 + 1$, p1 est supérieur à p2 et p3 et p4 tel que
       $p2 = 10x^5 + 20x^4 + 6x^2 + 6$, $p3 = 2x^7 + 40x$, $p4 = 2x^7 + 5x^2$

2. On veut construire une classe **PolyPuissance** qui représente la puissance entière d'un polynôme. exemple $(2x^4 + 5x^3 + 2)^3$. Pour cela, on étend la classe Polynôme. Expliciter la classe *PolyPuissance. De telle sorte que si la puissance de l'objet PolyPuisssance est négative une exception de type Exception est générée (l'exception n'est pas levée à ce niveau). Rajouter* uniquement les méthodes *évaluer(), toString() et* une méthode main qui crée un objet PolyPuissance avec le polynome $2x^7 + 40$ et la puissance n qui sera lue: Il faut lever l'exception et distinguer entre la valeur non entière `InputMismatchException` et la valeur entière négative Exception.

3. Répondre brièvement aux questions suivantes :
   Donner les relations (surcharge, redéfinition, …) entres toutes les méthodes ayant les mêmes noms dans les classes Monôme et Polynôme. Refaire le même travail entre les classes Polynôme et PolyPuissance.
   Expliquer brièvement comment le concept d'encapsulation a été respecté, quel est son intérêt.