

TD 2

Héritage

Exercice 1 : Promenade dans une hiérarchie d'héritage

On considère une hiérarchie constituée de trois classes, A, B et C, décrites partiellement ci-dessous.

```
public class A
{
    private int i;

    public A(int x) { i = x; }
    public String quiEsTu () {return "je suis un A";}
    public String toString() { return "i = " + i;}
    public String presenteToi() { return quiEsTu() + toString(); }
} // fin classe A

public class B extends A
{
    private int j;

    public B(int x, int y)
    { super(x);
      j = y;
    }
    public String quiEsTu () {return "je suis un B";}
    public String toString() {return super.toString() + "\n j = " + j;}
} // fin classe B

public class C extends B
{
    // aucun attribut supplémentaire
    public C(int x, int y) { super(x,y);}
    public String quiEsTu () {return "je suis un C";}
} // fin classe C
```

a- Détailler les actions effectuées par l'instruction suivante :

```
C obj = new C(5,10);
```

b- Quelles sont les méthodes exécutées lors de l'appel `obj.presenteToi()` ci-dessous ? Justifier (donner les principes qui expliquent, lors d'un appel, le choix de la méthode de telle ou telle classe).

```
C obj = new C(5,10);
System.out.println(obj.presenteToi());
```

c- Même question avec l'appel ci-dessous (où `obj` est de type A, tout en référant une instance de C) :

```
A obj = new C(5,10);
System.out.println(obj.presenteToi());
```

Exercice 2 : Une hiérarchie de pokemons

Les Pokémons sont des gentils animaux qui sont passionnés par la programmation objet en général et par le polymorphisme en particulier.

Il existe quatre grandes catégories de pokémons :

- **Les pokémons sportifs** : Ces pokémons sont caractérisés par un nom, un poids (en kg), un nombre de pattes, une taille (en mètres) et une fréquence cardiaque mesurée en nombre de pulsations à la minute. Ces pokémons se déplacent sur la terre à une certaine vitesse que l'on peut calculer grâce à la formule suivante : $vitesse = nombre\ de\ pattes * taille * 3$
- **Les pokémons casaniers** : Ces pokémons sont caractérisés par un nom, un poids (en kg), un nombre de pattes, une taille (en mètres) et le nombre d'heures par jour où ils regardent la télévision. Ces pokémons se déplacent également sur la terre à une certaine vitesse que l'on peut calculer grâce à la formule suivante : $vitesse = nombre\ de\ pattes * taille * 3$
- **Les pokémons des mers** : Ces pokémons sont caractérisés par un nom, un poids (en kg) et un nombre de nageoires. Ces pokémons ne se déplacent que dans la mer à une vitesse que l'on peut calculer grâce à la formule suivante : $vitesse = poids / 25 * nombre\ de\ nageoires$
- **Les pokémons de croisière** : Ces pokémons sont caractérisés par un nom, un poids (en kg) et un nombre de nageoires. Ces pokémons ne se déplacent que dans la mer à une vitesse que l'on peut calculer grâce à la formule suivante : $vitesse = (poids / 25 * nombre\ de\ nageoires) / 2$

Pour chacune de ces quatre catégories de pokémons, on désire disposer d'une méthode **toString** qui retourne (dans une chaîne de caractères) les caractéristiques du pokémon.

Par exemple la méthode toString appliquée sur un pokémon sportif retournerait :

"Je suis le pokémon Pikachu mon poids est de 18 kg, ma vitesse est de 5,1 km/h j'ai 2 pattes, ma taille est de 0,85m ma fréquence cardiaque est de 120 pulsations à la minute"

Cette même méthode appliquée sur un pokémon casanier pourrait retourner :

"Je suis le pokémon Salameche mon poids est de 12 kg, ma vitesse est de 3,9 km/h j'ai 2 pattes, ma taille est de 0,65m je regarde la télé 8h par jour"

Sur un pokémon des mers :

"Je suis le pokémon Rondoudou mon poids est de 45 kg, ma vitesse est de 3,6 km/h j'ai 2 nageoires"

Et enfin sur un pokémon de croisière :

"Je suis le pokémon Bulbizarre mon poids est de 15 kg, ma vitesse est de 0,9 km/h j'ai 3 nageoires"

Réaliser un diagramme de classes permettant de gérer toute la hiérarchie des Pokémons.

Cet exercice sera repris comme première étape du TP3.

Exercice 3 : Expressions arithmétiques

On considère l'évaluation d'expressions arithmétiques formées à l'aide des quatre opérateurs binaires +, -, *, /.

Une expression est définie récursivement de la façon suivante :

- soit c'est une *constante* (par exemple 1.5)
- soit c'est une *expression "complexe"* de la forme $(a\ op\ b)$ où a et b sont des *expressions* et op est l'un des quatre opérateurs.

Écrire les classes Java permettant de construire et évaluer des expressions, de façon à ce que l'on puisse écrire par exemple (et par exemple dans une méthode main appartenant à une autre classe) :

```
Constante a = new Constante (5);
Constante b = new Constante (2);
Constante c = new Constante(3);
ExpressionComplexe e1 = new ExpressionComplexe (a, '+', b);

ExpressionComplexe e2 = new ExpressionComplexe (e1, '*', c);
ExpressionComplexe e3 =
    new ExpressionComplexe(new Constante(4), '*', e2);

System.out.println(a.eval()); // 5.0
System.out.println(e1.eval()); // 7.0
System.out.println(e2.eval()); // 21.0
System.out.println(e3.eval()); // 84.0
```