

Exercice 1

On considère la classe `Relation` suivante, qui met en relation deux objets `a` et `b` :

```
public class Relation {
    private Object a;
    private Object b;
    public Relation(Object a, Object b) {
        this.a = a;
        this.b = b;
    }

    public Object a() {
        return a;
    }

    public Object b() {
        return b;
    }
}
```

Question 1 Modifier le code pour que la classe `Relation` soit paramétrée par un type `T` de façon que les deux objets `a` et `b` soient tous les deux de type `T`.

Question 2 Modifier le code pour que les deux objets `a` et `b` soient tous les deux d'un type `T` qui soit une sous-classe de la classe `Number` (comme `Integer`, `Float` ...).

Exercice 2

On considère l'interface `Evenement`. Pour simplifier, une date est codée par un entier :

```
public interface Evenement {
    int date();
    void changerDate(int date);
    String description();
    void changerDescription(String descr);
}
```

Question 1 Écrire la classe `Reunion` qui implémente `Evenement`. Une réunion sera de plus caractérisée par un lieu (`String`).

Exemple de code de test :

```
Evenement r = new Reunion(275, "seminaire", "Amphi_LaBRI");
```

Question 2 Modifier le code de `Reunion` pour que le code

```
Evenement r = new Reunion(275, "Seminaire", "Amphi_LaBRI");
System.out.println(r);
```

affiche

```
Seminaire, date : 275, lieu : Amphi Labri
```

Question 3 Modifier le code de `Reunion` pour que deux réunions soient égales si elles ont lieu le même jour et ont le même descriptif.

Quelle autre méthode faut-il redéfinir ? Donner le code correspondant.

Question 4 Modifier le code de `Reunion` pour que si l'on essaie de créer une réunion avec une date de valeur négative, une exception `DateNegativeException` soit levée. Donner également le code de `DateNegativeException`.

Question 5 On considère une autre classe, `Anniversaire` qui implémente `Evenement`. Un objet de cette classe est caractérisé par la date et le nom de la personne (description). Il n'est pas demandé d'écrire cette classe.

Comment factoriser le code de `Anniversaire` et `Reunion` ? Écrire le code factorisé et le nouveau code de `Reunion`.

Exercice 3

Qu'affiche le code suivant si on exécute la méthode `main()` de la classe `B` ?

```
class A {
    public String f() {
        return "A";
    }
}

class B extends A {
    private A a;

    public B(A a) {
        this.a = a;
    }

    public String f() {
        return super.f() + "B" + a.f();
    }

    public static void main(String[] args) {
        A a = new A();
        System.out.println(a.f());
        A a1 = new B(a);
        System.out.println(a1.f());
        A a2 = new B(a1);
        System.out.println(a2.f());
    }
}
```