

Durée du CC : 1h30.

La Fiche-Triche Scala est le seul document autorisé.

Nous allons représenter et manipuler des expressions sur les entiers positifs utilisant les opérations `+` (addition) et `min` (minimum entre deux entiers).

Question 1 Déclarez une classe abstraite `Expression` (destinée à représenter nos expressions), et scellez-là.

Déclarez ensuite les `case class` non-mutables suivantes, qui héritent de `Expression` :

- `Entier(n)`, où `n` est un entier positif; on vérifiera dans le constructeur que l'entier `n` est bien supérieur ou égal à zéro
- `Plus(e1,e2)`, où `e1` et `e2` sont des `Expression`
- `Min(e1,e2)` où `e1` et `e2` sont des `Expression`

Question 2 Implémentez la méthode `evaluer()` de la classe `Expression`. Celle-ci renverra un `Int` égal au résultat de l'expression. Pour calculer le minimum entre deux entiers, on utilisera la fonction `Math.min()`.

Question 3 Faites en sorte que l'égalité entre `Expression` vérifie si les deux expressions s'évaluent en la même valeur.

Question 4 Écrivez la méthode `simplifier()` de la classe `Expression`. Celle-ci renverra une nouvelle `Expression`, qui aura été simplifiée en appliquant les règles suivantes :

- `0` est l'élément neutre pour l'addition : autrement dit, `0+e` et `e+0` se simplifient en `e`,
- `0` est absorbant pour `min` : `min(0,e)` et `min(e,0)` se simplifient en `0`,

Question 5 En l'état, nos `Expression` ne peuvent pas être utilisées correctement dans des structures de données utilisant une fonction de hachage.

Rectifiez cette situation, en réimplémentant `hashCode()` de façon raisonnable.