

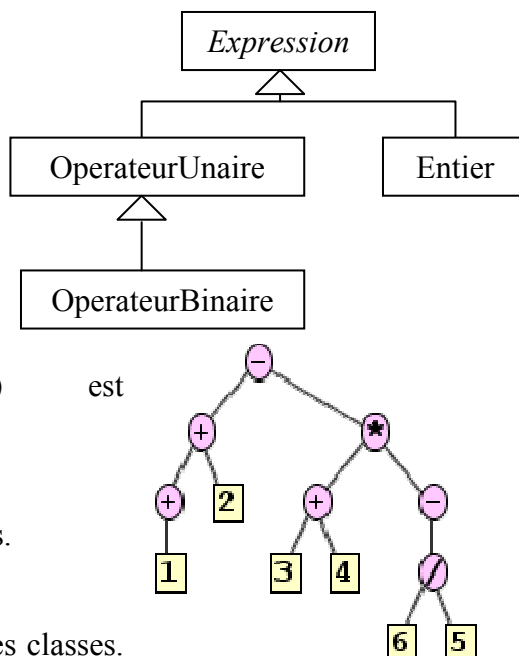
# Algorithmique et programmation

Nous nous proposons de représenter des expressions arithmétiques sur des entiers à l'aide des classes *Expression*, *Entier*, *OperateurUnaire* et *OperateurBinaire*. Les opérateurs pris en compte sont :

- +, - unaires.
- \*, / multiplication et division entière.
- +, - addition et soustraction binaires.

Exemple :

L'expression  $(+1+2) - (3+4) * - (6/5)$  est représentée par la structure :



**Question 1** (4 points)

Définir les constructeurs des différentes classes.

**Question 2** (4 points)

Définir la méthode *int evaluer()* des différentes classes.

La méthode *evaluer()* retourne la valeur d'une expression.

**Question 3** (3 points)

Définir la méthode *int getPriorite()* des différentes classes. La méthode *getPriorite()* retourne la priorité d'un opérateur. La priorité des opérateurs unaires est 3, la priorité de \* et / est 2, et la priorité du + et - binaires est 1. La priorité d'un *Entier* est 4.

**Question 4** (4 points)

Définir la méthode *String toString()* des différentes classes. La méthode *toString()* retourne une chaîne de caractères représentant l'expression en forme préfixée. Pour l'exemple précédent, la chaîne de caractères retournée est :

$(- (+ (+ 1) 2) (* (+ 3 4) (- (/ 6 5))))$

**Question 5** (5 points)

Définir la méthode *String toString2()* des différentes classes. La méthode *toString2()* retourne une chaîne de caractères représentant l'expression en forme infixée, avec seulement les parenthèses obligatoires. Pour l'exemple précédent, la chaîne de caractères retournée est :

$+1+2-(3+4)*-(6/5)$

```
abstract public class Expression {
    public Expression () {...}
    public abstract int evaluer();
    public abstract String toString();
    public abstract String toString2();
    public abstract int getPriorite();
}
```

```
class OperateurUnaire extends Expression {
    protected String op;
    protected Expression gauche;
    public OperateurUnaire(Expression g, String op){...}
    public int getPriorite(){...}
    public String toString(){...}
    public String toString2(){...}
    public int evaluer () {...}
}
```

```
class OperateurBinaire extends OperateurUnaire {
    protected Expression droit;
    public OperateurBinaire (Expression g, Expression d, String op){...}
    public int getPriorite(){...}
    public String toString(){...}
    public String toString2(){...}
    public int evaluer () {...}
}
```

```
class Entier extends Expression {
    protected int val;
    public Entier (int v){...}
    public int getPriorite(){...}
    public String toString(){...}
    public String toString2(){...}
    public int evaluer () {...}
}
```