

DÉVELOPPEMENT D'UNE CALCULATRICE



David Diema, Delrone Cherry

SOMMAIRE

- 1. Introduction
- 2. Répartition du Travail
- 3. Implémentation
- 4. Gestion des Exceptions
- 5. Difficultés rencontrées
- 6. Démonstration
- 7. Conclusion



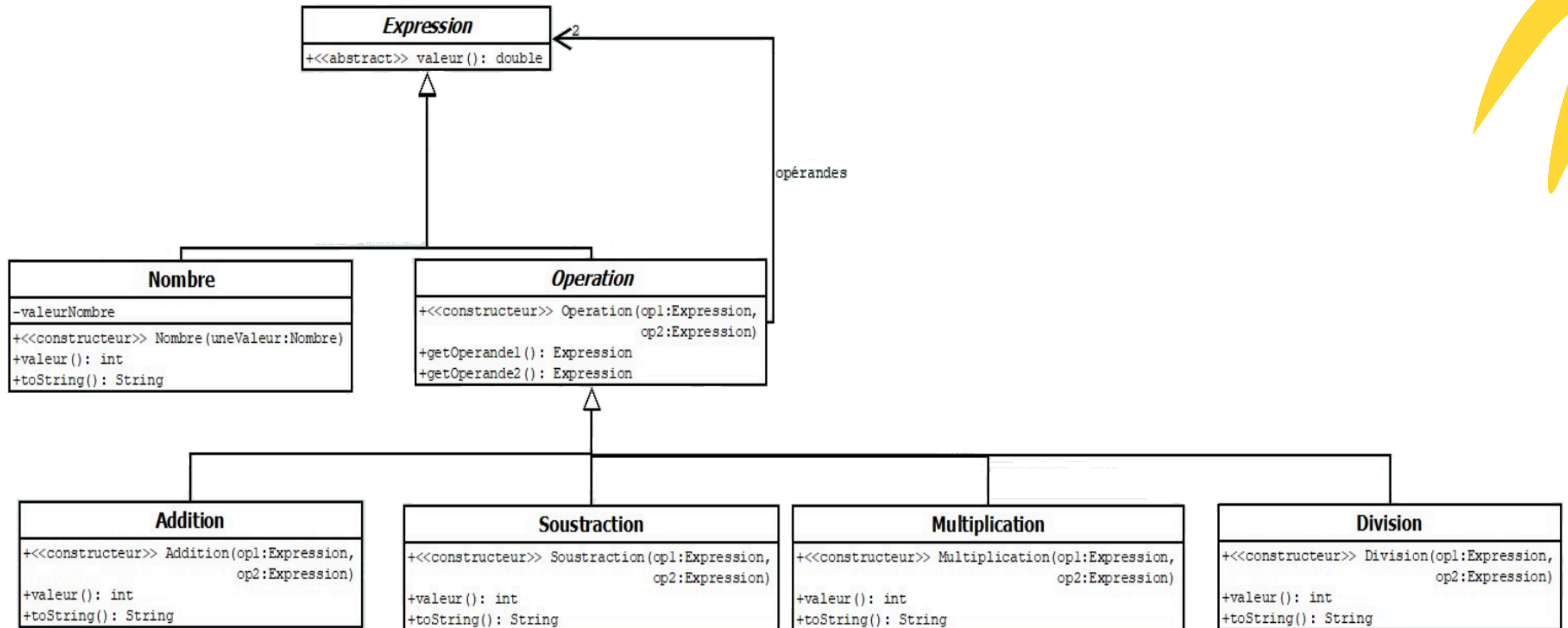
01

Notre équipe et nos objectifs

- Présentation de l'équipe
- Objectif du projet 



Répartition du Travail



Implémentation

Présentation des différentes classe et nos choix.

```
1 public class Multiplication extends Operation{
2
3     public Multiplication(Expression operande1, Expression operande2){
4         super(operande1, operande2);
5     }
6
7     public double valeur(){
8         return getOperande1().valeur()* getOperande2().valeur();
9     }
10
11     public String toString() {
12         return getOperande1() + " * " + getOperande2();
13     }
14 }
15
16
```

```
1 public abstract class Expression {
2     public abstract double valeur();
3 }
```

```
1 public abstract class Operation extends Expression {
2     private Expression operande1;
3     private Expression operande2;
4
5     public Operation(Expression operande1, Expression operande2) {
6         this.operande1 = operande1;
7         this.operande2 = operande2;
8     }
9
10    public abstract double valeur();
11
12    public Expression getOperande1(){
13        return this.operande1;
14    }
15    public Expression getOperande2(){
16        return this.operande2;
17    }
18
19 }
```



Gestion des Exceptions

```
1 public class Division extends Operation{
2
3     public Division(Expression operande1, Expression operande2){
4         super(operande1, operande2);
5         if(operande2.valeur()==0){
6             throw new IllegalArgumentException("Division par 0 impossible");
7         }
8     }
9
10    public double valeur(){
11        return getOperande1().valeur()/getOperande2().valeur();
12    }
13
14    public String toString(){
15        return getOperande1().toString()+ "/" + getOperande2().toString();
16    }
17 }
```

Difficultés rencontrées

- Gestion la nouvelle classe Expression
- Compréhension de la méthode valeur()
- Signification de la classe Opération
- Appréhension vis à vis du Throw



Démonstration

```
1 public class Calculatrice{
2     public static void main(String[] args) {
3         try {
4             Expression deux = new Nombre(2) ;
5             Expression trois = new Nombre(3) ;
6             Expression dixSept = new Nombre(17) ;
7             Expression s = new Soustraction(dixSept, deux) ;
8             Expression a = new Addition(deux, trois) ;
9             Expression d = new Division(s, a) ;
10            System.out.println(d + " = " + d.valeur());
11        } catch (Exception e) {
12            System.out.println("Il y a une erreur : " + e.getMessage());
13        }
14    }
15 }
```


Conclusion

Retour sur notre expérience.