



Présentation de notre SAE Java

Lucas DJAVID - Zakarya MANAR

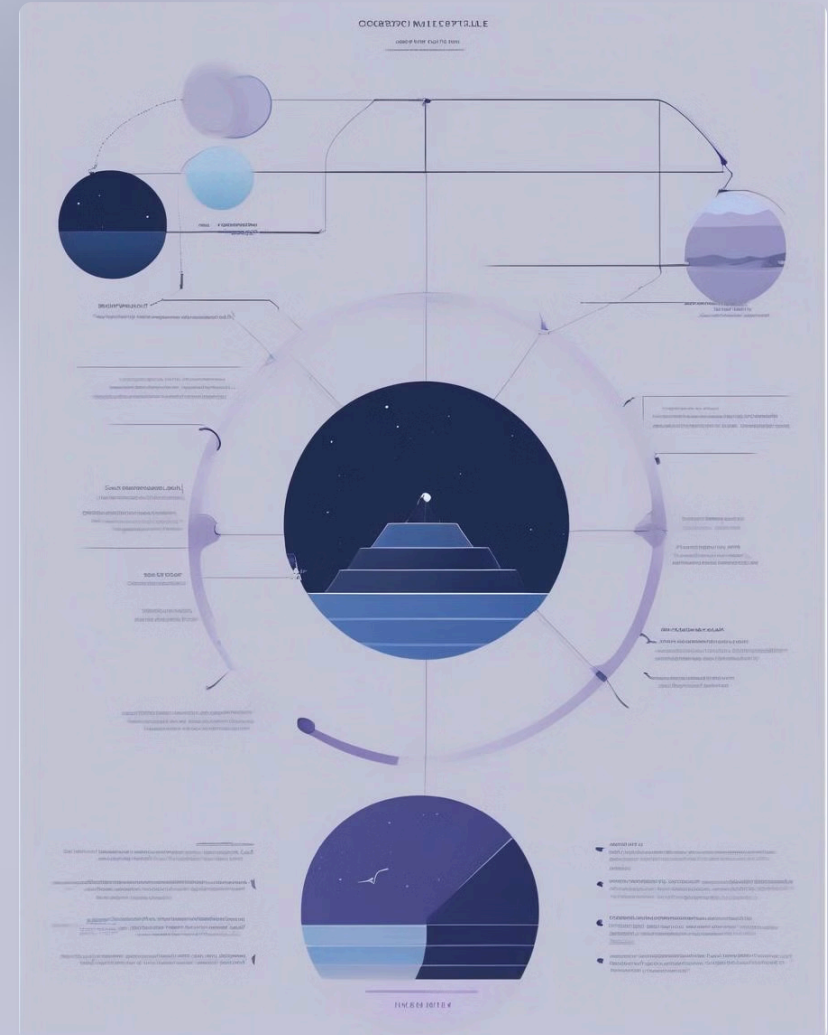
Bienvenue à la présentation de notre projet Java. Ce projet a impliqué une collaboration d'équipe, une conception méticuleuse et une mise en œuvre efficace. Plongeons-nous dans les étapes clés qui ont mené au succès de ce projet.

Conception du projet

Comprendre les exigences du projet, identifier les fonctionnalités clés, et définir les objectifs.

Concevoir le projet en utilisant un diagramme UML. Ce diagramme permet de visualiser les différentes classes, leurs relations et le fonctionnement général du système.

Définir les interactions de l'utilisateur avec le système, en créant des interfaces intuitives et conviviales.



Organisation de notre binôme

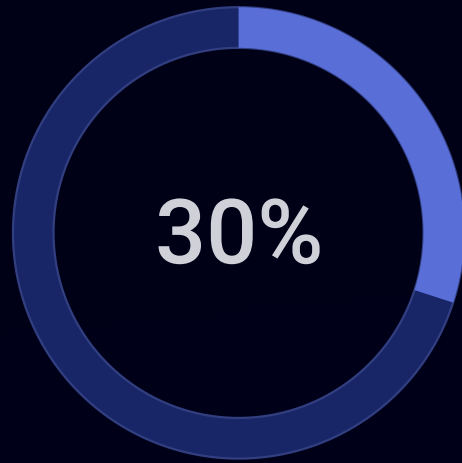
Lucas DJAVID

- Réalisation du Diagramme UML
- Classes Addition, Soustraction et Multiplication

Zakarya MANAR

- Classes Division, Nombre, Opération, et Expression
- Gestion des erreurs

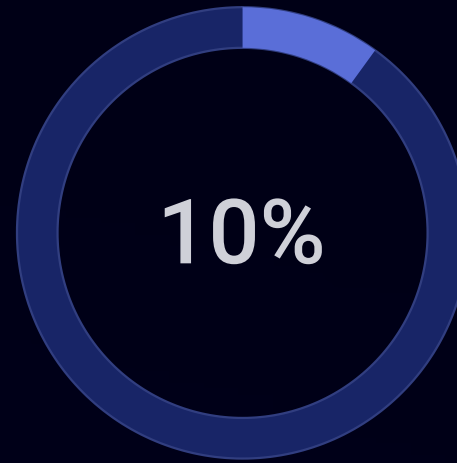
Difficultés rencontrées



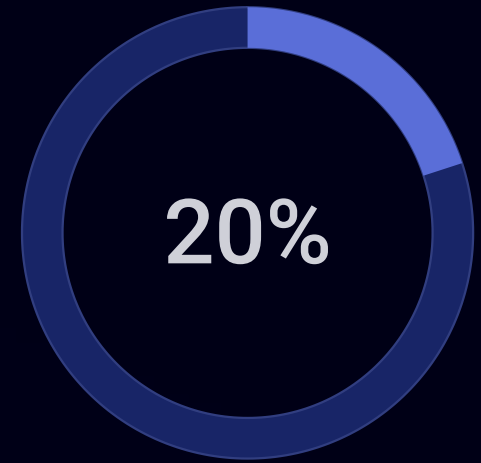
Détectations
d'erreurs



Résolution
d'erreurs

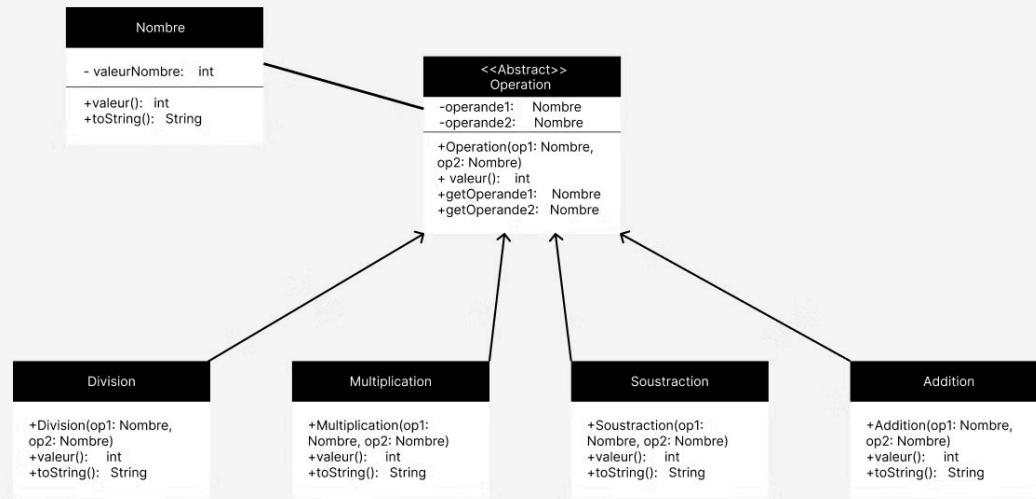


Répartition des
tâches



Compréhension
des consignes

Diagramme UML



Problème: Infinity

```
(9 / 0) = Infinity
```

```
(9 / (6 - (2 * 3))) = Infinity
```

Problème Infinity

Infinity apparaît lors de la division par 0.0

1

2

Cause

Une variable utilisée comme diviseur prenait la valeur 0.0, provoquant un calcul incorrect et renvoie Infinity.

Solution

Ajouter une condition dans le code pour vérifier si le diviseur est égal à 0.0, si tel est le cas un `throw Exception` renvoie l'erreur "Division par zéro".

3

4

Leçon apprise

Il est important de traiter les exceptions de façon précise.

Erreur de type

```
.\Nombre.java:1: error: Nombre is not abstract and does not override abstract method valeur() in Expression
public class Nombre extends Expression {
    ^
.\Nombre.java:8: error: valeur() in Nombre cannot override valeur() in Expression
    public int valeur() {
        |         ^
    return type int is not compatible with double
```

Erreur de type

Une "erreur de type" se produit lorsqu'il y a une incompatibilité entre les types de données utilisés dans le programme.

1

2

Cause

L'erreur de type est due au fait que la classe expression utilise comme méthode "valeur()" de type double, or on ne peut pas redéfinir cette méthode dans nombre en changeant le type par int malgré que l'on manipule que des entiers.

Solution

Garder le type double lors de la redéfinition de méthode pour éviter cette erreur.

3

4

Leçon apprise

Démonstration en direct du code

1

Présentation du code

Explication du code Java en temps réel, en mettant en évidence les points clés et les concepts utilisés.

2

Exécution du code

Démonstration du fonctionnement du code, en montrant comment l'application s'exécute et en mettant en avant ses fonctionnalités principales.

3

Réponses aux questions

Si vous avez des questions sur le code nous sommes aptes à y répondre.



Compétences acquises

Objectifs	Clés de réussite
Définir les besoins du projet	Communication efficace entre les membres de l'équipe.
Travailler en collaboration	Respect des rôles et des responsabilités de chaque membre.
Gérer les erreurs	Test et débogage réguliers du code.

Conclusion

1

Travailler en collaboration

Coordonner les efforts de l'équipe de manière efficace

2

Gérer les erreurs

Identifier et résoudre rapidement les problèmes

3

Atteindre les objectifs

Livrer un projet réussi et satisfaire les attentes

Questions et réponses

