

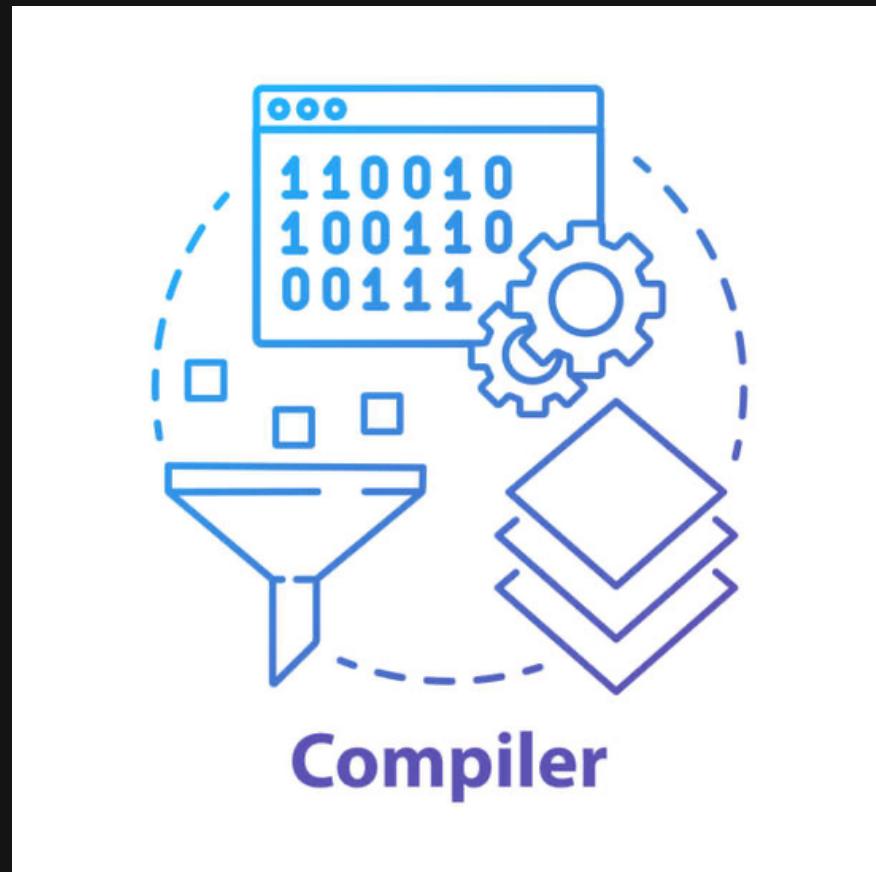
# Compilateur Decac

Besoin d'un compilateur zero défaut ?

GL35



# Sommaire :



- Qu'est-ce qu'un Compilateur ?
- Produit : Options et utilisation.
- Les messages d'erreur.
- Architecture.
- Limitations du produit.
- Extension : Classe Math.
- Validation du produit.
- Demonstration.
- Bilan du projet.

# Qu'est-ce qu'un Compilateur ?

## Code source

```
{  
    print("Hello world ")  
}
```

## Code Objet

```
WSTR "Hello world"  
HALT
```

## Execution

```
>_
```

```
Hello world
```

# Le Produit :

Date de sortie : 23 janvier 2023

- Produit moderne.
- Labelisé Ensimag.
- Eco Friendly.

# Les Options

COMPILEUR DECAC

Decac

Decac -b

Decac -r

Decac -p

Decac -v

Decac -P

Decac -n

# L'utilisation :

Compilateur DECAC

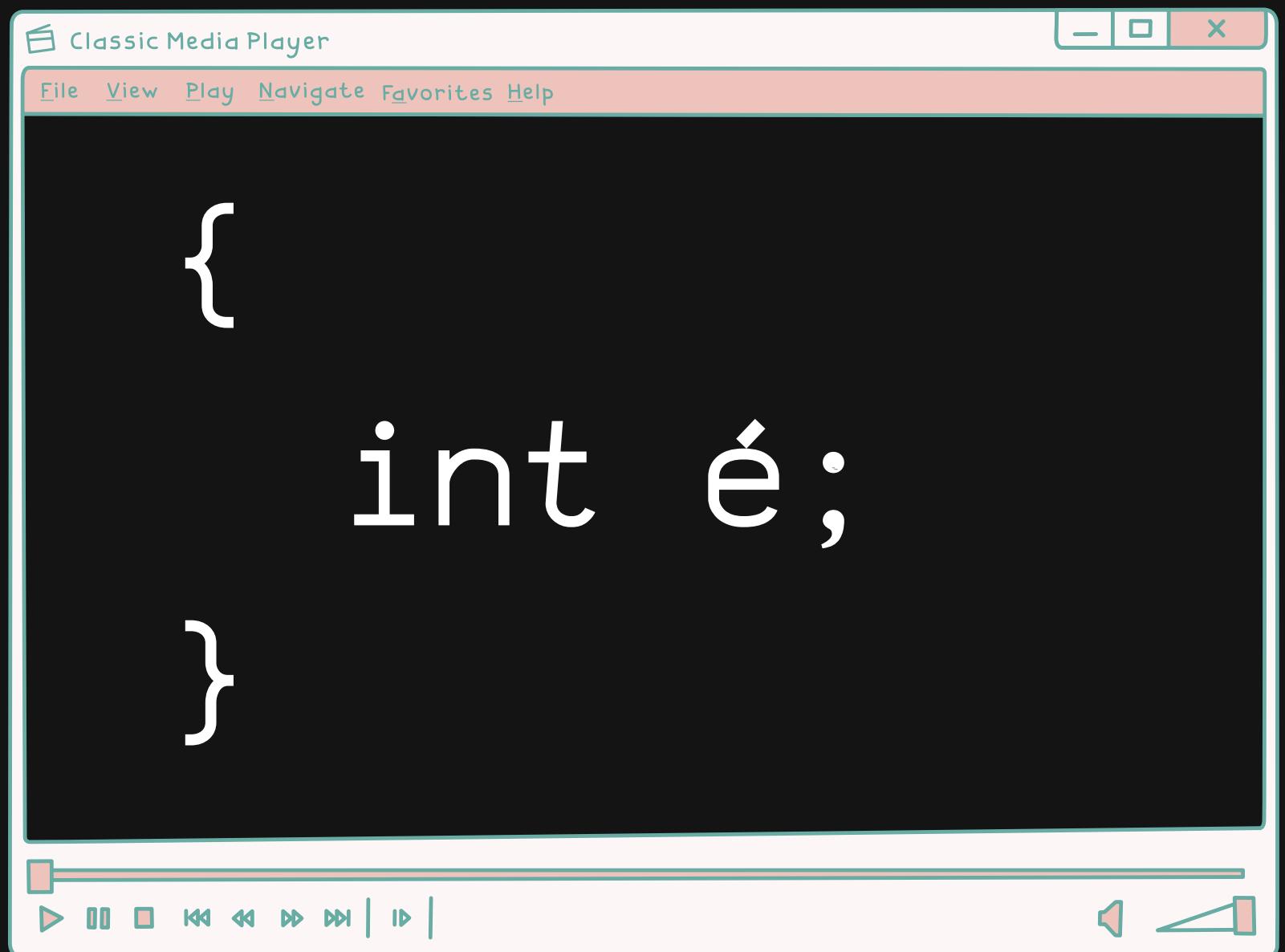
UTILISATION SIMPLE ET GUIDÉE  
PAR UN MANUEL D'UTILISATEUR  
TRÈS DETAILLÉ.

DES MESSAGES D'ERREURS  
COMPREHENSIBLE (TYPE  
D'ERREURS ET LIGNE ...)

EFFICACE POUR CODER DES  
SOLUTIONS AUX PROBLEMES  
REELS (RECURSIVITE,  
ORIENTEE OBJET ...)

# LES MESSAGES D'ERREUR :

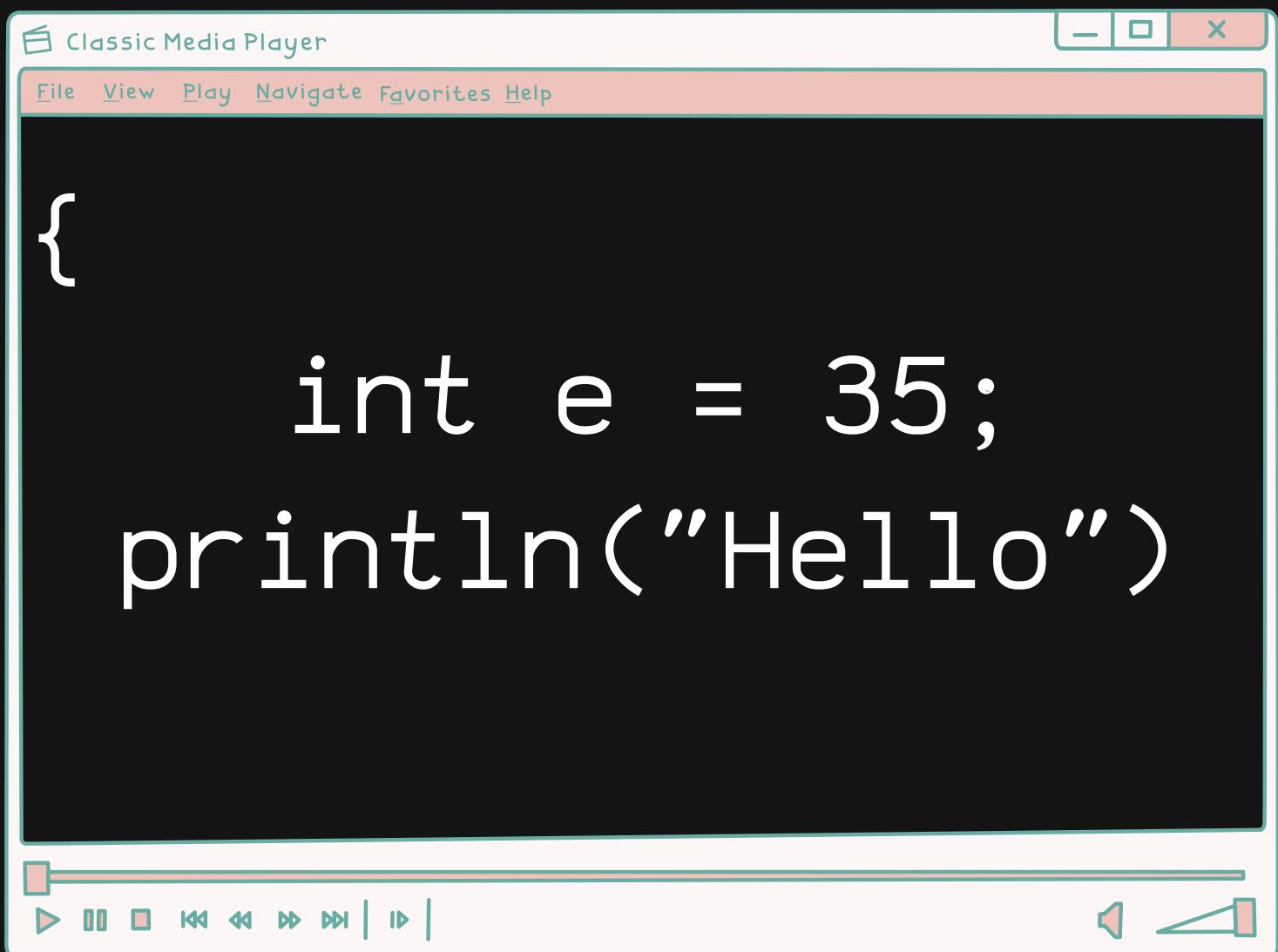
## Analyse lexicale



accent.deca:2:5: token recognition error at: 'é'

# LES MESSAGES D'ERREUR :

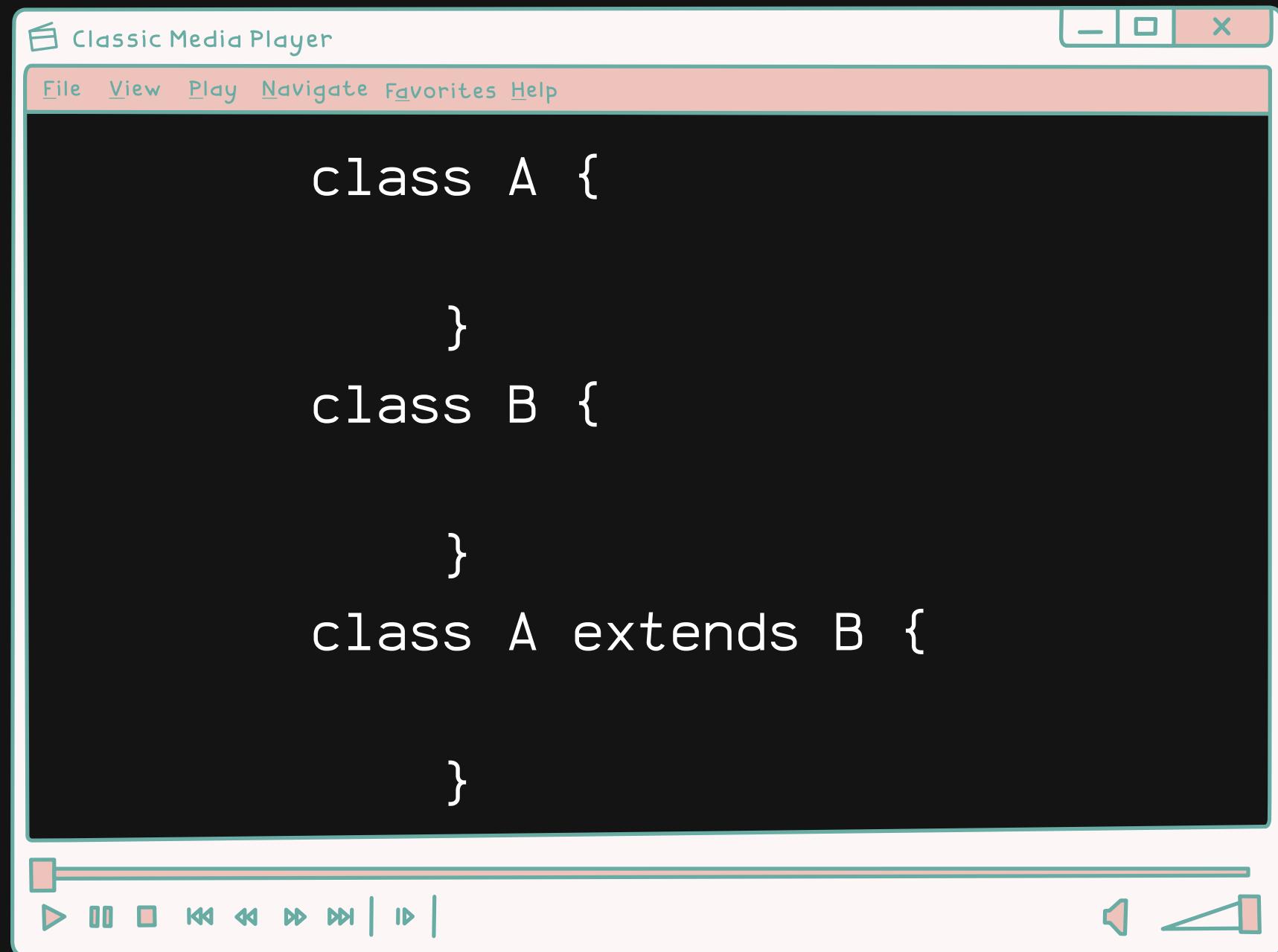
## Analyse syntaxique



```
blocnotclosed.deca:5:0: mismatched input '<EOF>' expecting
{'instanceof', '<', '>', '=', '==', '!=', '<=', '>=', '+', '-',
'*', '/', '%', '.', ',', ')', '|', '||', '&&'}
```

# LES MESSAGES D'ERREUR :

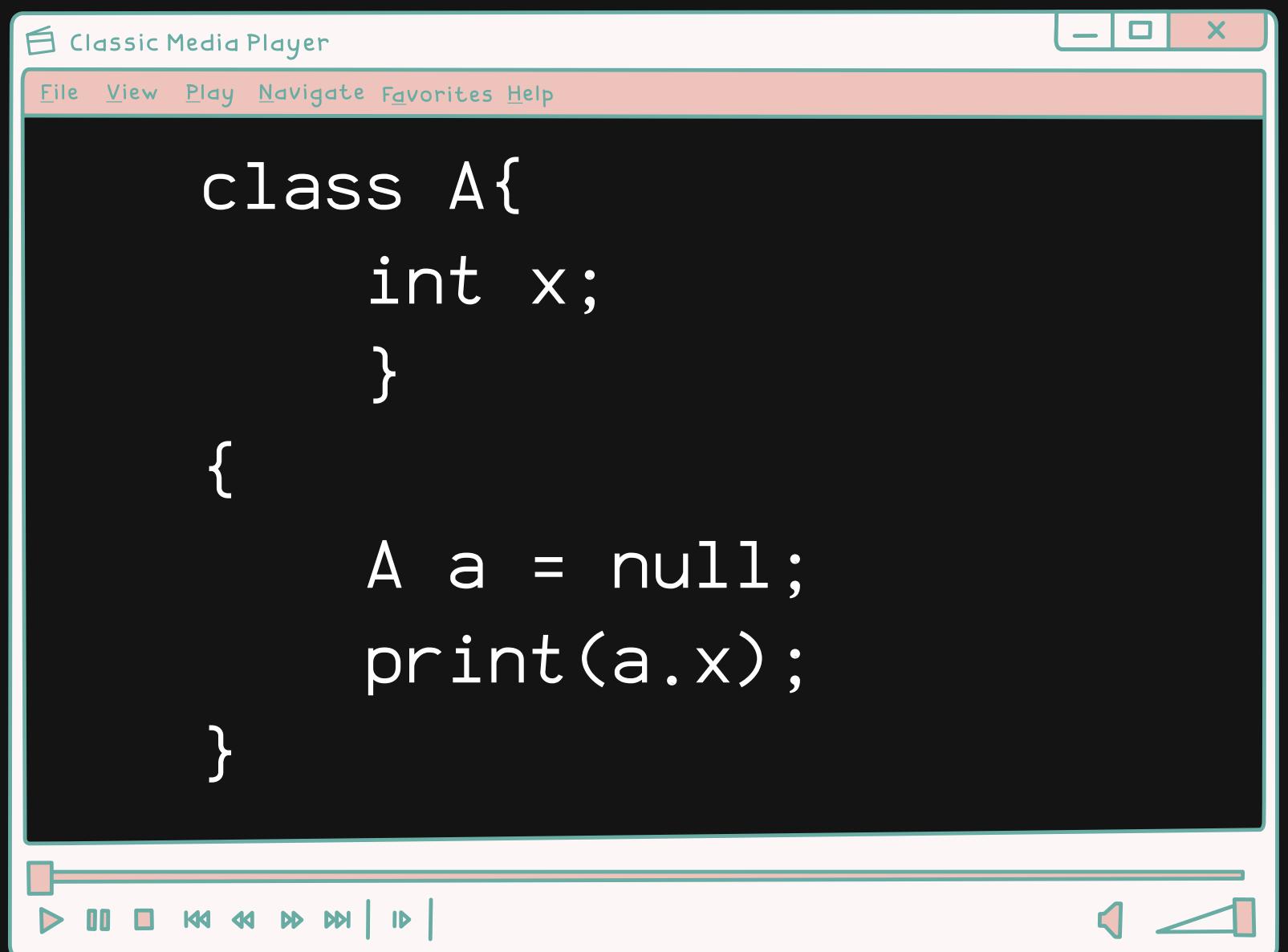
## Analyse contextuelle



doubledefclass.deca:7:0: The class already exists. (1.3)

# LES MESSAGES D'ERREUR :

## Generation de code



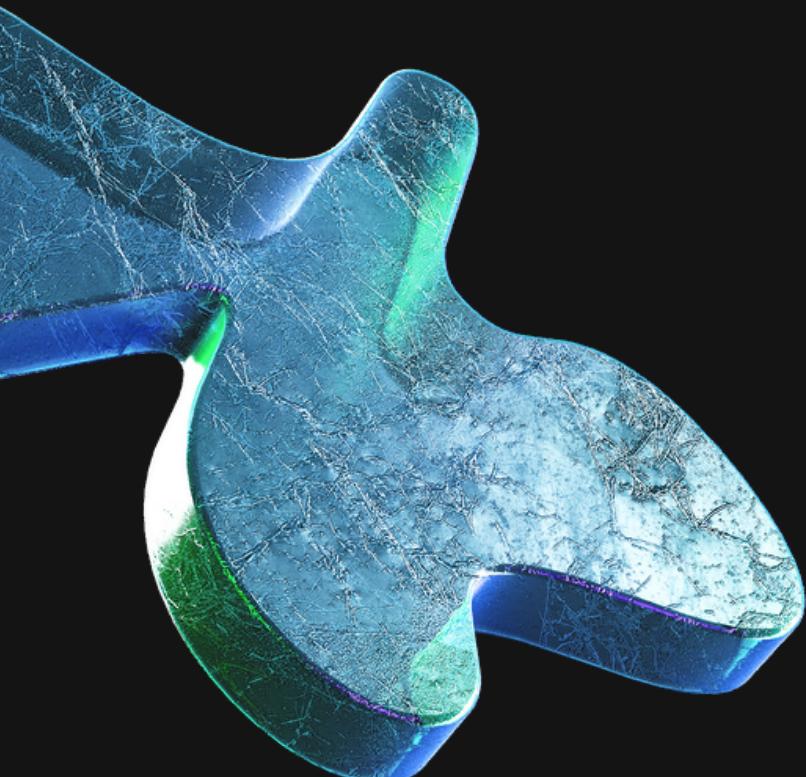
The image shows a window titled "Classic Media Player" with a dark theme. The menu bar includes "File", "View", "Play", "Navigate", "Favorites", and "Help". The main area contains the following Java-like code:

```
class A{  
    int x;  
}  
  
{  
    A a = null;  
    print(a.x);  
}
```

The code has several syntax errors: "print(a.x);" is missing a closing parenthesis, and "print" is misspelled as "print". The window also features a media control bar at the bottom with icons for play, pause, stop, and volume.

Erreur : dereferencement de null

# Architecture et limitations



# Architecture de votre compilateur

ANALYSE LEXICALE ET  
SYNTAXIQUE

---

VERIFICATION  
CONTEXTUELLE

---

GÉNÉRATION DE  
CODE

---

# Analyse lexicographique

---

- Vérifier que tous les mots correspondent au langage deca.
- L'inclusion des fichiers si nécessaire.



# Analyse syntaxique

---

- Vérifier le respect de la grammaire.
- La construction de l'arbre syntaxique abstraite.

# Vérification contextuelle et décoration de l'arbre

PREMIER PASSE: VÉRIFICATION  
DES DÉCLARATIONS DE CLASSES

DEUXIÈME PASSE: VÉRIFICATION  
DES CHAMPS ET MÉTHODES

TROISIÈME PASSE: VÉRIFICATION  
DU CORPS DES CLASSES ET  
MÉTHODES

# Génération de code

## PREMIÈRE PASSE: CRÉATION DE LA TABLE DES MÉTHODES

Elle consiste à la construction du tableau des étiquettes des méthodes. Et la génération du code permettant la construction de celle-ci.

## DEUXIEME PASSE: CODAGE DES DIFFÉRENTES COMPOSANTES

Codage des champs, des méthodes, et du Main (Programme principal).

# Limitations du compilateur

GESTION DES  
DÉBORDEMENTS POUR  
LES ENTIERS,  
FLOTTANTS ET STRINGS

---

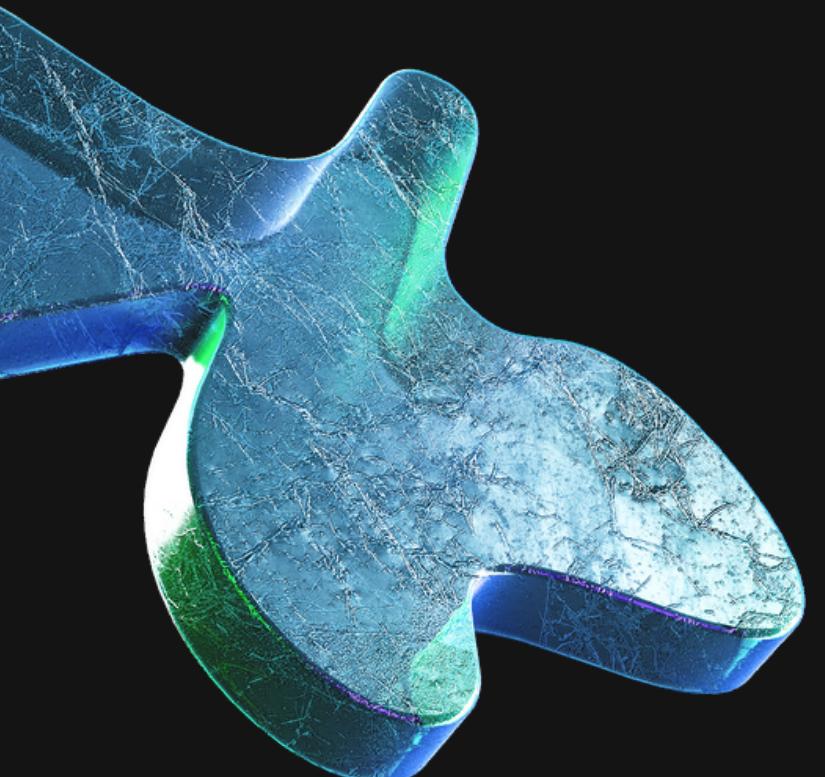
DÉBORDEMENTS DE LA  
PILE OU DU TAS

---

INCOMPATIBILITÉ DES  
MÉTHODES  
ASSEMBLEURS DE IMA

---

# Extension: Trigo



# Class Maths

## FONCTIONS IMPLÉMENTÉES

Ulps, Sin, Cos, Asin, Atan

---

## CHOIX D'IMPLÉMENTATION

Séries de Taylor

Polynomes d'Hermites pour Atan

---

## PRÉCISION

Ne dépasse pas les 4 ULP prés

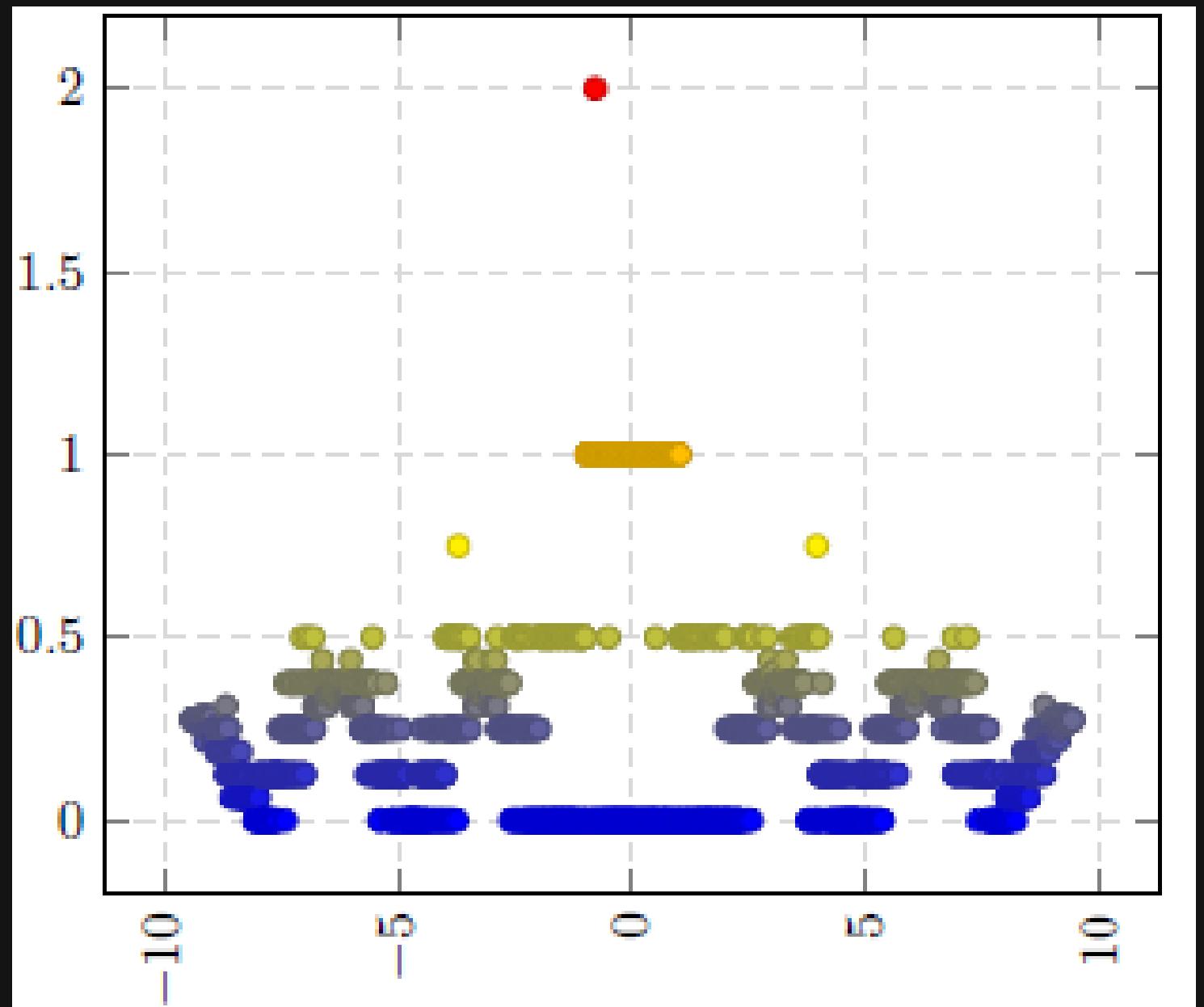
# RÉSULTATS

## ULP

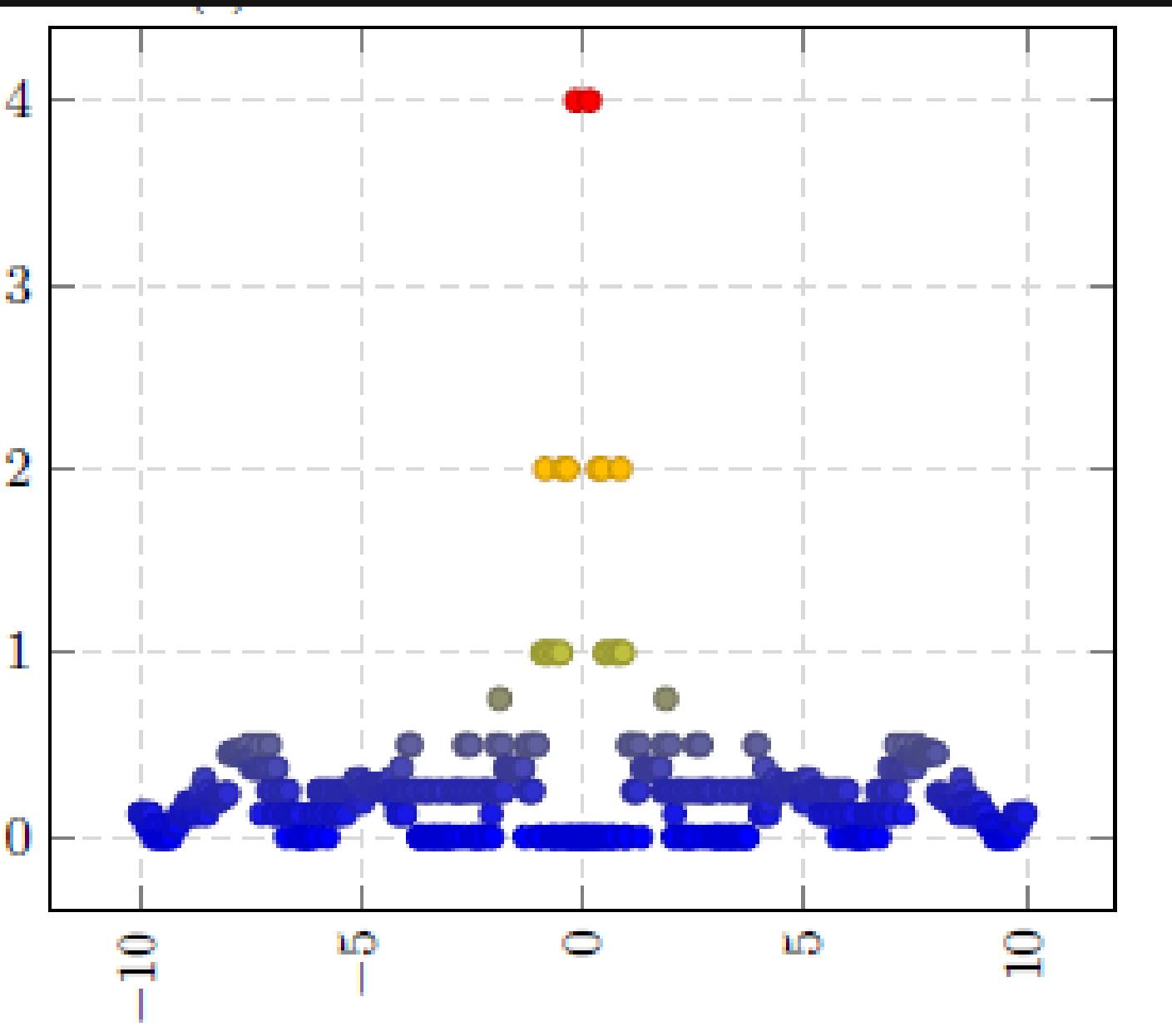
Intervalle	Pas	Nombre d'échantillons	Erreur entre notre ulp et Math.ulp
$[0; 10]$	$3.8146973 * 10^{-6}$	2621440	0
$[2^{-40}; 2^{-7}]$	$2^{-28}$	2097152	0
$[2^{-7}; 2^{12}]$	$9.765625 * 10^{-4}$	4194296	0
$[2^{12}; 2^{40}]$	$9.765625 * 10^{-4}$	4194296	0

# RÉSULTATS

SIN

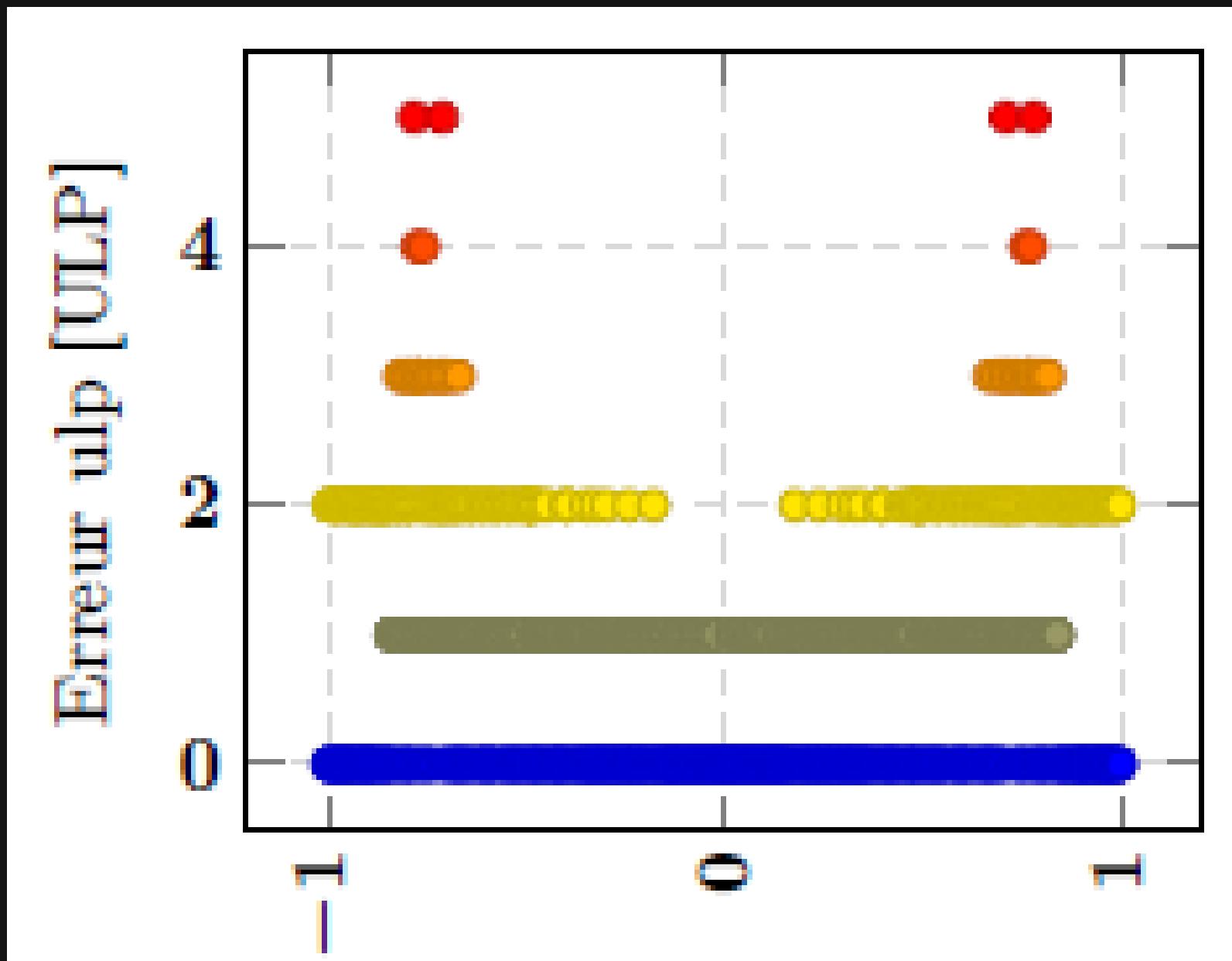


COS

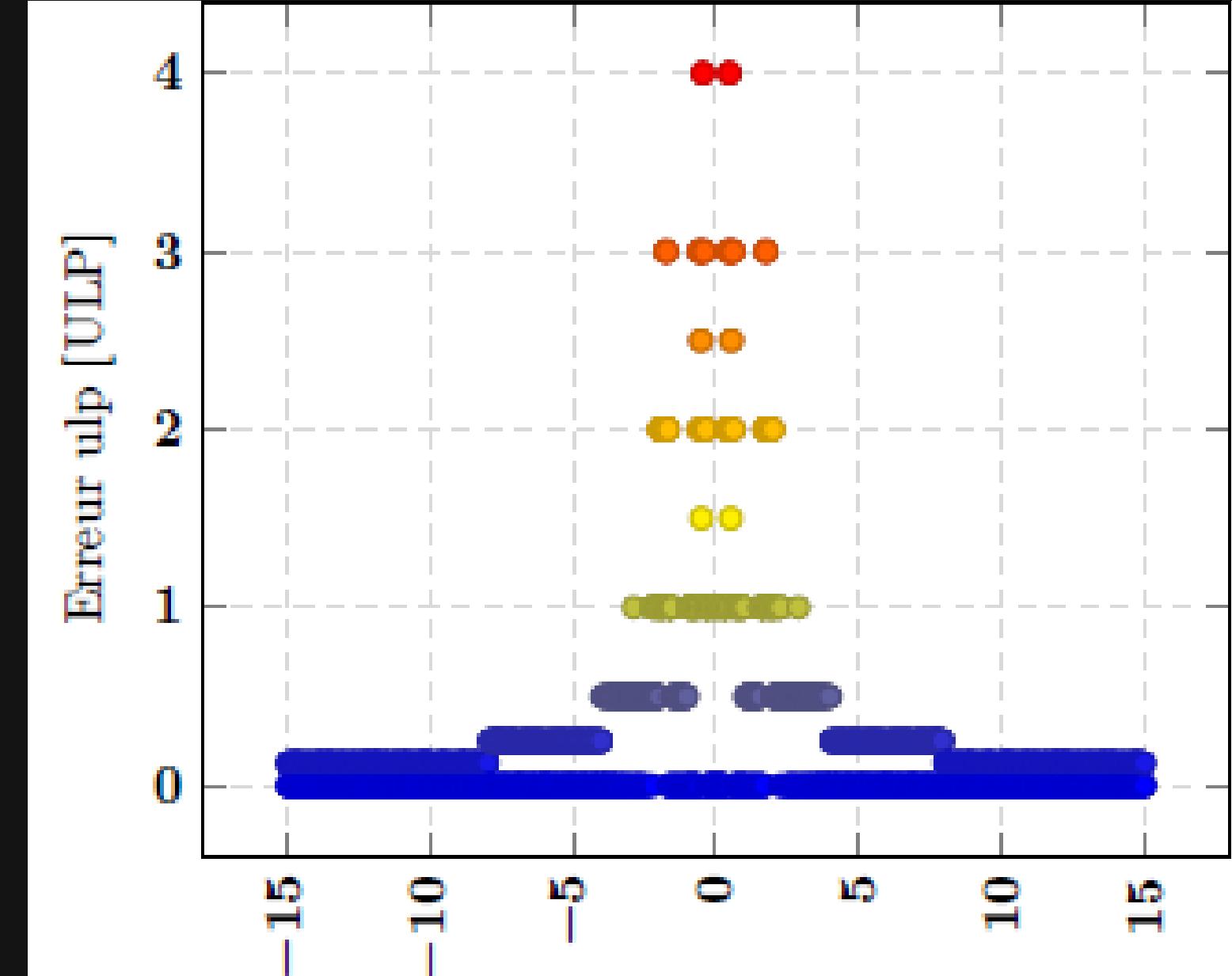


# RÉSULTATS

ASIN



ATAN



# QUELQUES VALEURS SUR DECA

```
cos(0.0)= 1.00000e+00  
0x1p+0  
valeur qu'il faut trouver calculé par java:0x1p+0  
-----  
cos(pi/4)= 7.07107e-01  
0x1.6a09e6p-1  
valeur qu'il faut trouver calculé par java:0x1.6a09e6p-1  
-----  
cos(pi/8)= 9.23880e-01  
0x1.d906bcp-1  
valeur qu'il faut trouver calculé par java:0x1.d906bcp-1  
-----  
cos(pi/12)= 9.65926e-01  
0x1.ee8dd4p-1  
valeur qu'il faut trouver calculé par java:0x1.ee8dd4p-1  
-----  
cos(pi/16)= 9.80785e-01  
0x1.f6297ep-1  
valeur qu'il faut trouver calculé par java:0x1.f6297cp-1  
cos(20)= 4.08083e-01  
0x1.a1e084p-2  
valeur qu'il faut trouver calculé par java:0x1.a1e044p-2
```

```
asin(0.5)= 5.23599e-01  
0x1.0c1522p-1  
valeur a trouver0x1.0c1524p-1  
-----  
asin(0.7)= 7.75397e-01  
0x1.8d00e6p-1  
valeur a trouver0x1.8d00e6p-1  
-----  
asin(0.1)= 1.00167e-01  
0x1.9a492ap-4  
valeur a trouver0x1.9a4928p-4  
-----  
asin(0.3)= 3.04693e-01  
0x1.38015ap-2  
valeur a trouver0x1.38015ap-2  
-----  
asin(1)= 1.57080e+00  
0x1.921fb6p+0  
valeur a trouver0x1.921fb6p+0
```

# QUELQUES VALEURS SUR DECA

```
atan(-1)= -7.85398e-01  
-0x1.921fb6p-1  
la valeur reelle: -0x1.921fb6p-1
```

```
-----  
atan(0.1)= 9.96687e-02  
0x1.983e28p-4  
la valeur reelle: 0x1.983e28p-4
```

```
-----  
atan(0.3)= 2.91457e-01  
0x1.2a73a4p-2  
la valeur reelle: 0x1.2a73a6p-2
```

```
-----  
atan(0.032)= 3.19891e-02  
0x1.060dfap-5  
la valeur reelle: 0x1.060df6p-5
```

```
-----  
atan(0,005)= 4.99996e-03  
0x1.47ad62p-8  
la valeur reelle: 0x1.47ad62p-8  
la valeur reelle: 0x1.47ad62p-8
```

```
atan(0.6)= 5.40420e-01  
0x1.14b1dep-1  
la valeur reelle: 0x1.14b1dep-1
```

```
-----  
atan(0.7)= 6.10726e-01  
0x1.38b112p-1  
la valeur reelle: 0x1.38b112p-1
```

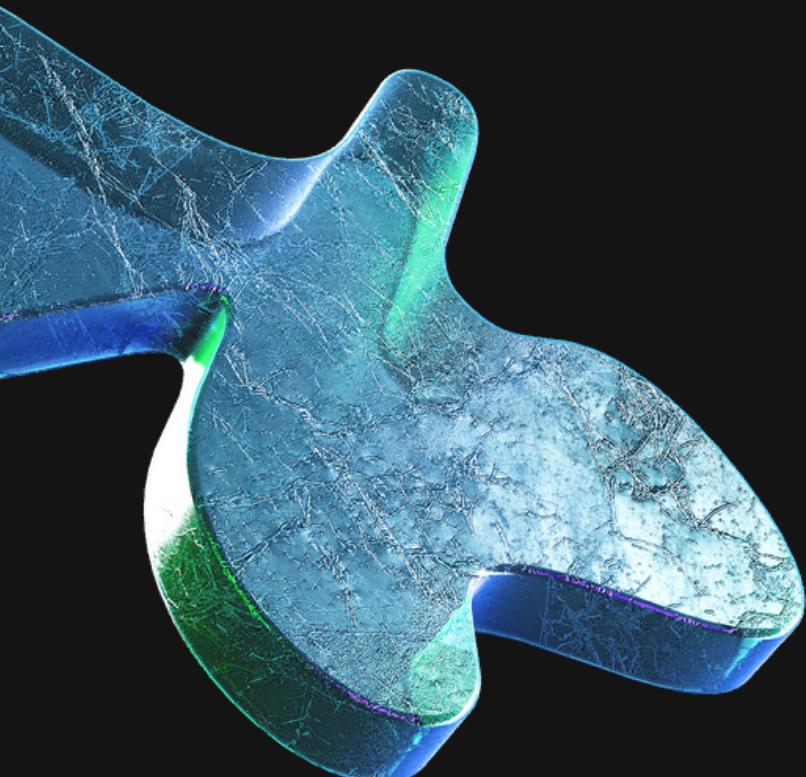
```
-----  
atan(20)= 1.52084e+00  
0x1.8555a4p+0  
la valeur reelle: 0x1.8555a2p+0
```

```
-----  
atan(0,0001)= 1.00000e-04  
0x1.a36e2ep-14  
la valeur reelle: 0x1.a36e26p-14  
la valeur reelle: 0x1.a36e26p-14
```

```
-----  
atan(5656)= 1.57062e+00  
0x1.92142p+0  
la valeur reelle: 0x1.92141ep+0
```

```
-----  
atan(0.0000009)= 9.00000e-07  
0x1.e32f0ep-21  
la valeur reelle: 0x1.e32f0ep-21
```

# Validation de notre Compilateur



# L'analyseur syntaxique et lexicale

## Étape A

### TESTS VALID

---

vérifier l'arbre abstrait de programme.

### TESTS INVALID

---

vérifier les messages d'erreurs

### DÉCOMPILEATION

---

vérifier le programme deca généré

# Analyse Contextuelle

## Étape B

### TESTS VALID

---

vérifier l'arbre décoré (type, définition...)

### TESTS INVALID

---

vérifier les messages d'erreurs (Violation des règles)

# Génération de code Assembleur

## Étape C

### TESTS VALID

---

vérifier les opérations arithmétiques , booléennes, tableau de méthodes...

### TESTS INVALID

---

vérifier les messages d'erreurs à l'exécution.

### INTERACTIVE

---

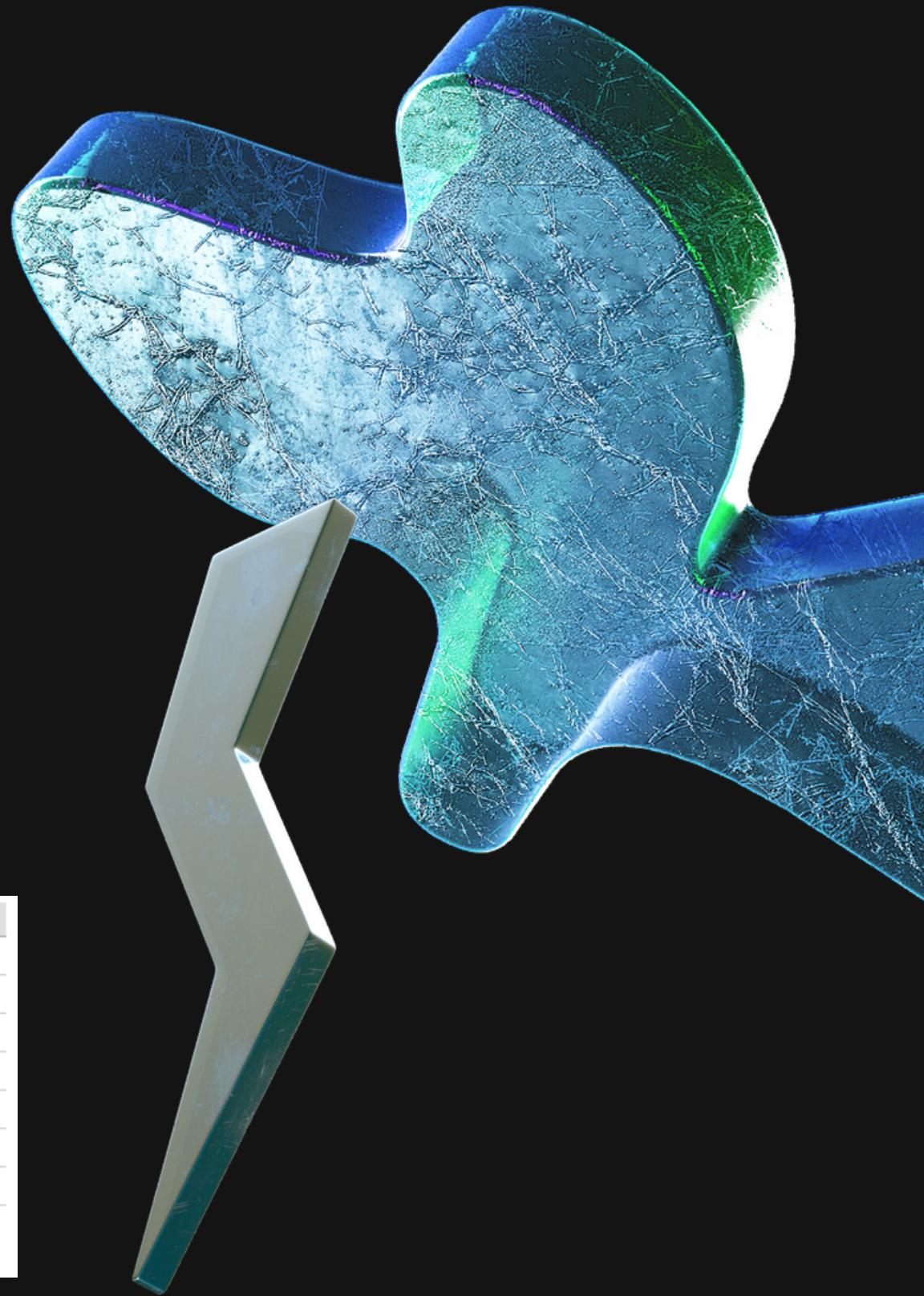
vérifier les fonctions Readint et Readfloat.

# La couverture des Tests

Jacoco

Une couverture de 82% (+230 tests)

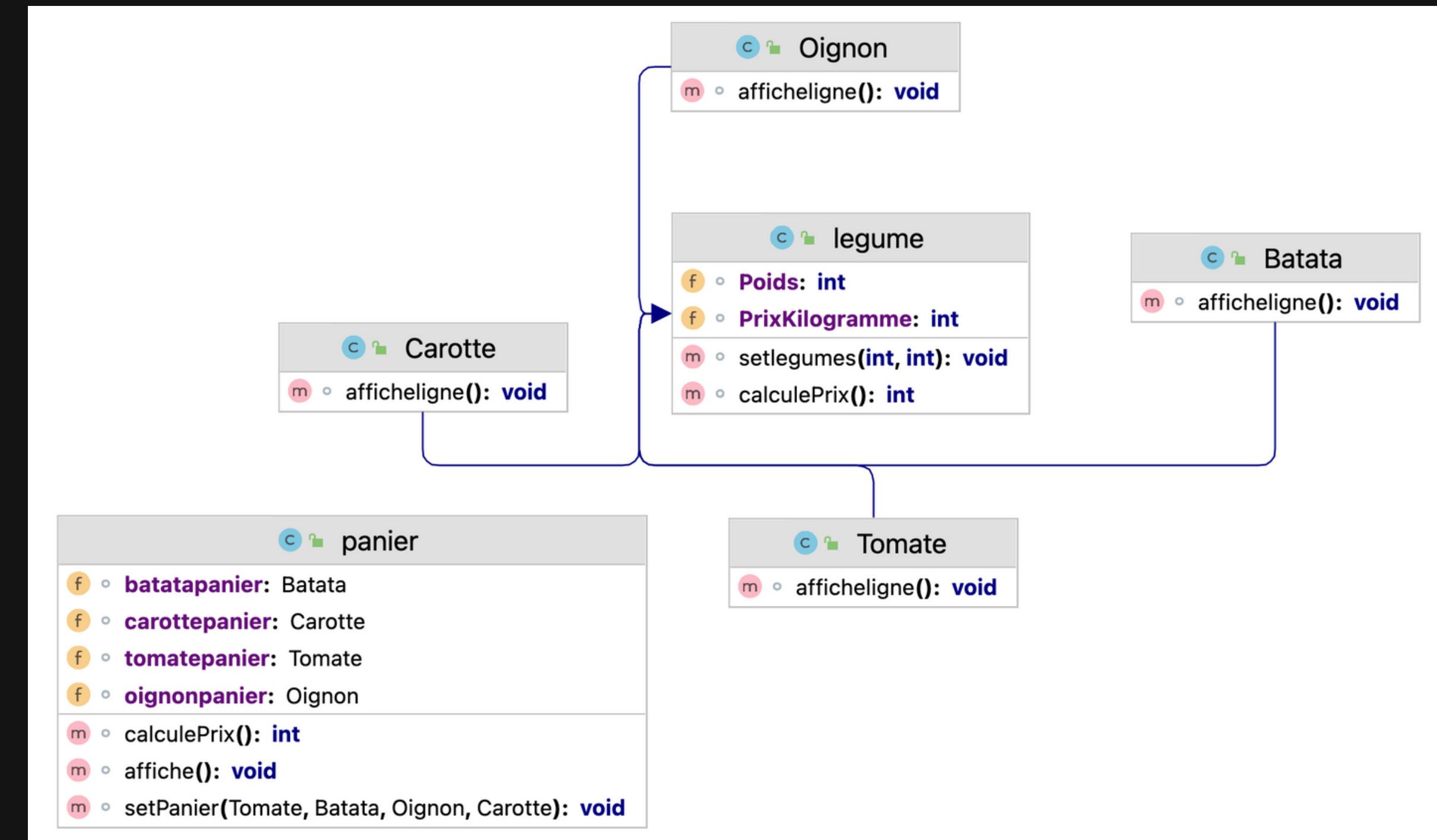
Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
<a href="#">fr.ensimag.deca.syntax</a>	<div style="width: 24%; background-color: red;"></div> <div style="width: 76%; background-color: green;"></div>	76 %	<div style="width: 18%; background-color: red;"></div> <div style="width: 82%; background-color: green;"></div>	58 %	451	661	463	1 971	241	368	0	49
<a href="#">fr.ensimag.deca.tree</a>	<div style="width: 11%; background-color: red;"></div> <div style="width: 89%; background-color: green;"></div>	89 %	<div style="width: 14%; background-color: red;"></div> <div style="width: 84%; background-color: green;"></div>	84 %	129	769	215	1 989	61	505	0	85
<a href="#">fr.ensimag.deca</a>	<div style="width: 3%; background-color: red;"></div> <div style="width: 77%; background-color: green;"></div>	77 %	<div style="width: 2%; background-color: red;"></div> <div style="width: 69%; background-color: green;"></div>	69 %	26	87	53	259	5	50	1	6
<a href="#">fr.ensimag.ima.pseudocode</a>	<div style="width: 1%; background-color: red;"></div> <div style="width: 81%; background-color: green;"></div>	81 %	<div style="width: 1%; background-color: red;"></div> <div style="width: 81%; background-color: green;"></div>	81 %	23	86	32	185	19	75	2	26
<a href="#">fr.ensimag.deca.context</a>	<div style="width: 1%; background-color: red;"></div> <div style="width: 90%; background-color: green;"></div>	90 %	<div style="width: 1%; background-color: red;"></div> <div style="width: 80%; background-color: green;"></div>	80 %	32	141	26	235	24	118	0	22
<a href="#">fr.ensimag.ima.pseudocode.instructions</a>	<div style="width: 0%; background-color: red;"></div> <div style="width: 81%; background-color: green;"></div>	81 %		n/a	13	62	22	111	13	62	10	54
<a href="#">fr.ensimag.deca.tools</a>	<div style="width: 0%; background-color: red;"></div> <div style="width: 94%; background-color: green;"></div>	94 %	<div style="width: 0%; background-color: red;"></div> <div style="width: 100%; background-color: green;"></div>	100 %	1	16	3	39	1	13	0	3
Total	3 891 of 22 393	82 %	341 of 1 205	71 %	675	1 822	814	4 789	364	1 191	13	245



# Démonstration



Un programme qui calcule le  
prix d'un panier des légumes  
et affiche le ticket de caisse  
correspondant.

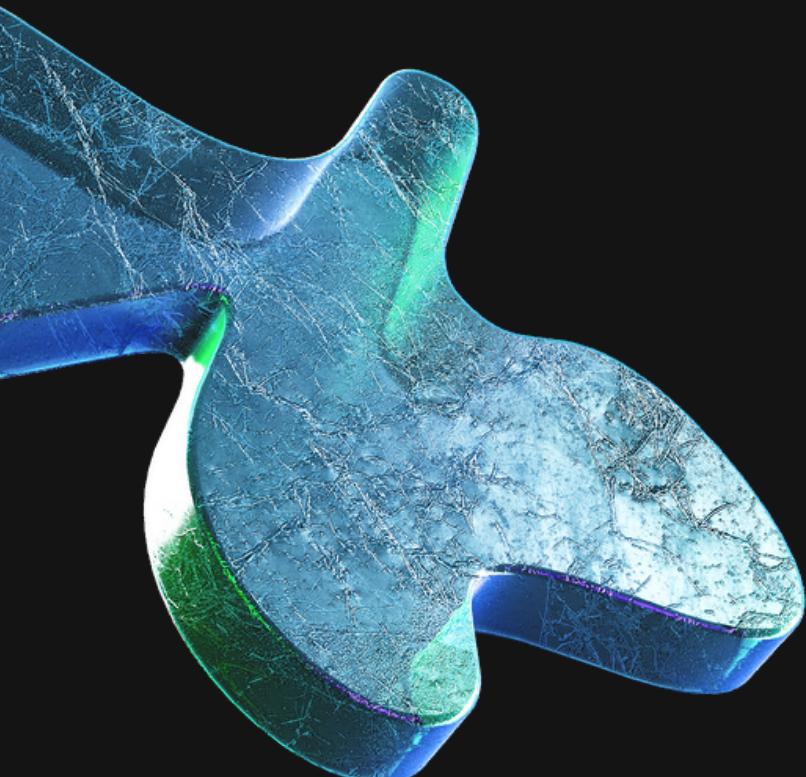


# Programme Main

```
{  
    panier MonPanier = new panier();  
    Tomate tomate = new Tomate();  
    Batata batata = new Batata();  
    Oignon oignon = new Oignon();  
    Carotte carotte = new Carotte();  
    tomate.setlegumes(4,10);  
    batata.setlegumes(3,11);  
    oignon.setlegumes(6,3);  
    carotte.setlegumes(7,2);  
    MonPanier.setPanier(tomate,batata,oignon,carotte);  
    MonPanier.affiche();  
    println();  
    println("PrixTotal: " ,MonPanier.calculePrix());  
}
```

```
morad@morad-HP-Pavilion-Gaming-Laptop-15-dk1xxx:~/ensimag/GL/gl35/src/test/deca/codegen/valid/improvided$ ima gestionPanier.ass  
Produit  PrixKg  Poids  
Tomate   4       10  
Batata   3       11  
Oignon   6       3  
Carotte  7       2  
  
PrixTotal: 105
```

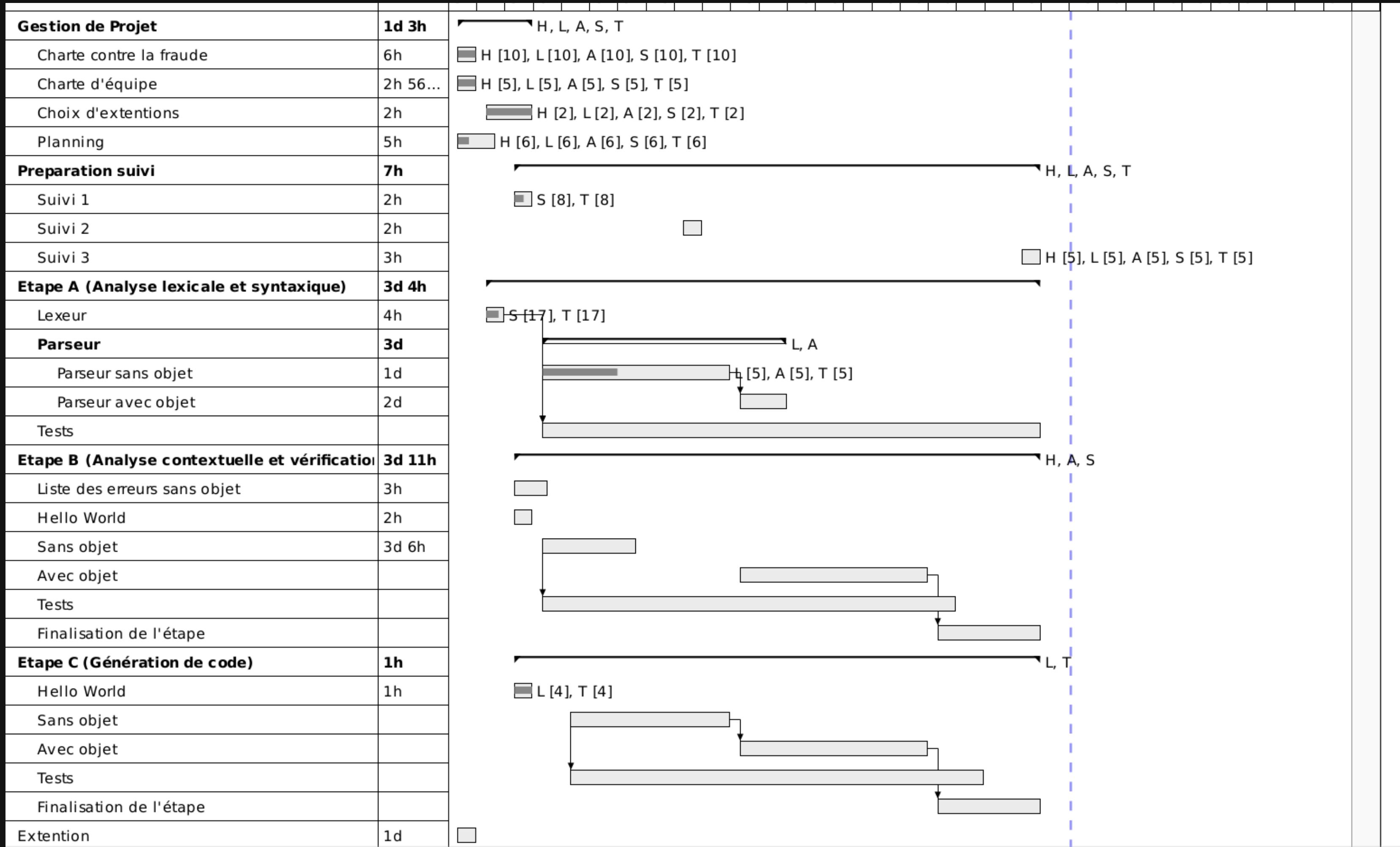
# Bilan de projet



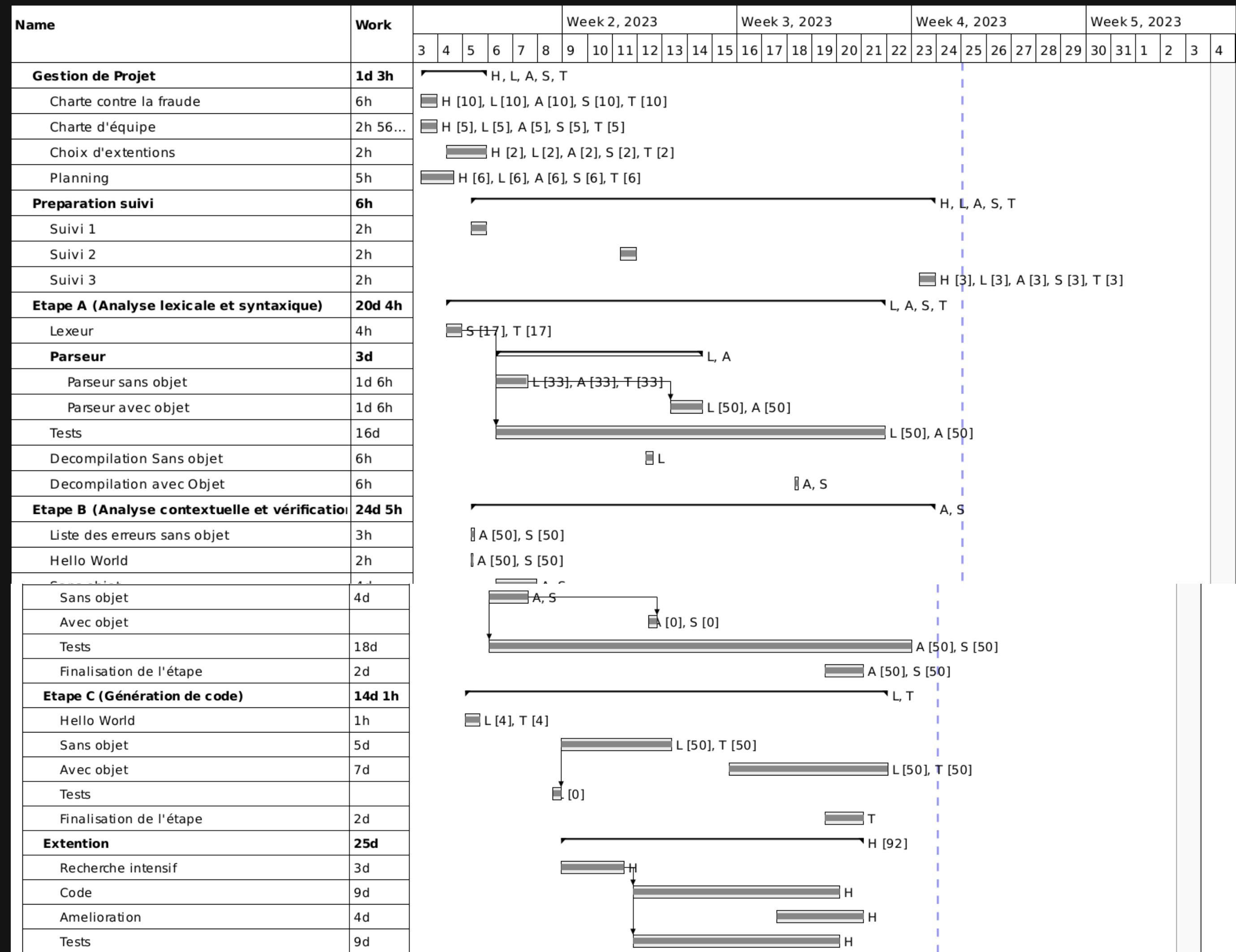
## ORGANISATION

- RÉUNION QUOTIDIENNE
- TRAVAILLE EN GROUPE
- DÉVELOPPEMENT AGILE

# PLANNING



# RÉALISATION



## RÉPARTITION DES TÂCHES

Membres	Rôles
Mohamed Tifaf	Scrum Master, Developer(A, C)
Sana Ailla	Developer(A, B), Tester (B, C)
Morad Laglil	Developer(A, C), Tester (A, C)
Yassin Habib	Developer(Extension), Tester (A, C)
Mohamed-Aymane Akil	Developer(A, B), Tester (B, A)

# RÉPARTITION DES TÂCHES

Nom	Documentation	Etape A	Etape B	Etape C	Extension	Tests	Total Result
Ailla	18	8	64	0	0	58	148
Akil	18	32	64	0	0	34	148
Habib	18	0	0	0	104	36	158
Laglil	18	32	0	30	0	72	152
Tifaf	18	24	0	104	0	16	162
<b>Total Result</b>	<b>90</b>	<b>96</b>	<b>128</b>	<b>134</b>	<b>104</b>	<b>216</b>	<b>768</b>

## DIFFICULTÉS RENCONTRÉS

- EVALUATION DES TÂCHES À RÉALISER
- SYNCHRONISATION ENTRE LES ÉTAPES

# RÉTROSPECTIVE

## GLAD

- GESTION EFFICACE DES DEADLINES ET DES RENDUS INTERMÉDIAIRES
- HORAIRES DE TRAVAIL COMPATIBLES AVEC TOUS LES MEMBRES DE L'ÉQUIPE

## SAD

- AMÉLIORER LA STRATÉGIE DE GESTION DES TESTS

## MAD

- DÉTÉRIORATION DE LA QUALITÉ DE VIE EN PÉRIODE DE PROJET

MERCI POUR VOTRE ATTENTION