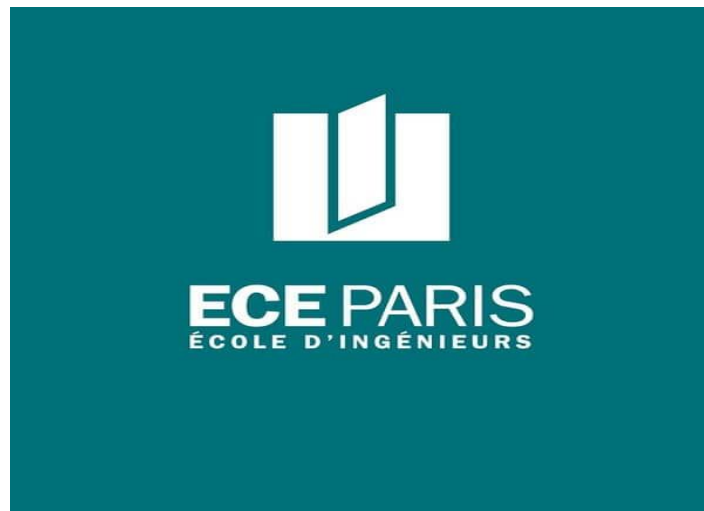


# Rapport TP2- Groupe TD4

---

DIAGRAMME DE CLASSES - CLASSES ET OBJETS –  
VISIBILITE - CAST – ENCAPSULATION – HERITAGE –  
PACKAGES

---



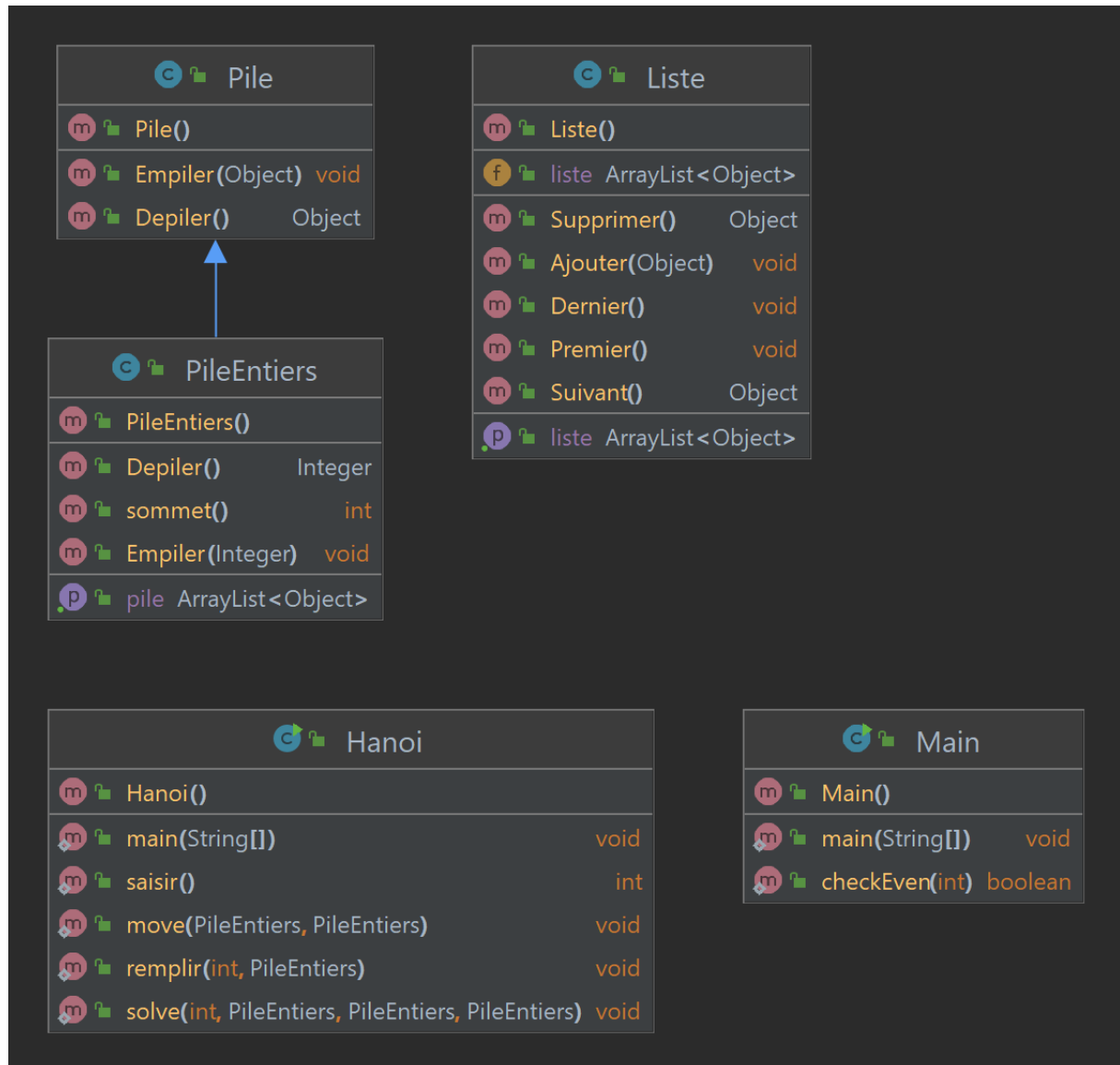
## Membres du Groupe :

- MBOSSO MBOSSO Guy
- KENMOE POLA François

## Table des matières

1/ Diagramme de classes .....	3
2/ Jeux d'essais des exécutable .....	4
a- exécutable de la question 3 .....	4
b- exécutable de la question 4 .....	5
3/ Bilan et conclusion personnalisé .....	6
4/ Bibliographie .....	7

# 1/ Diagramme de classes



## 2/ Jeux d'essais des exécutables

a- exécutable de la question 3

```
C:\Users\sadow\.jdk\openjdk-19.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\
Etape 0
P1 : [-437, 441, -621, -107, 330]
P2 : []
P3 : [-437]

Etape 1
P1 : [-437, 441, -621, -107, 330]
P2 : []
P3 : [-437, 441]

Etape 2
P1 : [-437, 441, -621, -107, 330]
P2 : []
P3 : [-437, 441, -621]

Etape 3
P1 : [-437, 441, -621, -107, 330]
P2 : []
P3 : [-437, 441, -621, -107]
```

```
Etape 3
P1 : [-437, 441, -621, -107, 330]
P2 : []
P3 : [-437, 441, -621, -107]

Etape 4
P1 : [-437, 441, -621, -107, 330]
P2 : [330]
P3 : [-437, 441, -621, -107]

Process finished with exit code 0
```

A la question 3, le programme remplit la pile d'entiers P1 avec cinq entiers relatifs aléatoirement choisis dans l'intervalle d'entiers [-1000 ; 1000]. Puis il remplit les piles P2 et P3 avec les nombres pairs et impairs de P1 respectivement.

## b- exécutable de la question 4

```
Run: Hanoi <
C:\Users\sedow\.jdk\openjdk-19.0.1\bin\java.exe "-javaagent:C:\Pro
entrer le nombre de disque
3
etape 1
T1 =[3, 2, 1]
T2 =[ ]
T3 =[ ]

T1 =[3, 2]
T2 =[ ]
T3 =[1]

T1 =[3]
T2 =[2]
T3 =[1]

T1 =[3]
T2 =[2, 1]
T3 =[ ]

etape 2
T1 =[3]
T2 =[2, 1]
T3 =[ ]
```

```
Run: Hanoi <
T3 =[ ]

etape 2
T1 =[3]
T2 =[2, 1]
T3 =[ ]

T1 =[ ]
T2 =[2, 1]
T3 =[3]

T1 =[1]
T2 =[2]
T3 =[3]

T1 =[1]
T2 =[ ]
T3 =[3, 2]

etape 3
T1 =[1]
T2 =[ ]
T3 =[3, 2]
```

```
Run: Hanoi <
T1 =[1]
T2 =[ ]
T3 =[3, 2]

etape 3
T1 =[1]
T2 =[ ]
T3 =[3, 2]

T1 =[ ]
T2 =[ ]
T3 =[3, 2, 1]
```

Le programme effectue la résolution du problème des tours de Hanoï en fonction du nombre  $n$  de disques se trouvant initialement sur la première tour. Dans ce cas particulier  $n=3$ , la résolution est optimale.

### 3/ Bilan et conclusion personnalisé

En somme, ce TP nous a permis d'améliorer nos compétences en programmation orienté objet et de maîtriser des algorithmes de résolution du problème des tours de Hanoï.

Guy : << J'ai mieux compris les notions de package et d'héritage. Ma plus grande difficulté a été l'implémentation de la méthode permettant la résolution du problème en affichant les différentes étapes>>

#### 4/ Bibliographie

[https://fr.wikipedia.org/wiki/Tours de Hano%C3%AF](https://fr.wikipedia.org/wiki/Tours_de_Hano%C3%AF)