



**UNIVERSITÉ  
DE REIMS  
CHAMPAGNE-ARDENNE**

**TP2**

INFO0502

11 novembre 2024

VERPOORTE William

## Sommaire

|          |                      |          |
|----------|----------------------|----------|
| <b>1</b> | <b>Objectif</b>      | <b>2</b> |
| <b>2</b> | <b>Diagramme UML</b> | <b>2</b> |
| <b>3</b> | <b>Commentaire</b>   | <b>2</b> |
| <b>4</b> | <b>main</b>          | <b>2</b> |
| <b>5</b> | <b>Texas Hold'em</b> | <b>7</b> |

# 1 Objectif

Nous voulons mettre en place une version simple du Poker et à la fin du Poker Texas Hold'em, qui va implémenté la création de carte, de main, du talon d'où sont tirés les cartes. Ensuite nous simulons un jeu entre 4 joueurs et déduisons le vainqueur grâce au différentes combinaisons possibles de cartes.

## 2 Diagramme UML

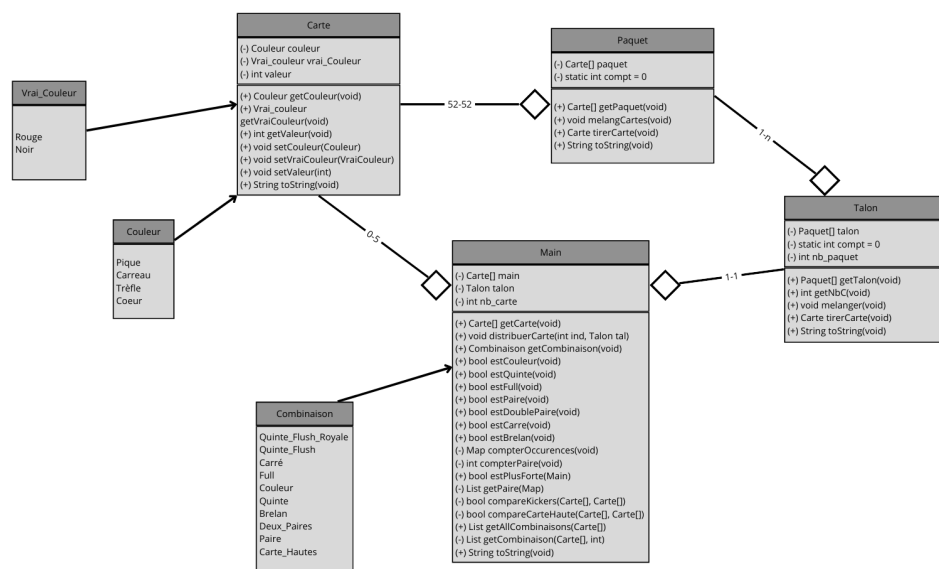


Figure 1: Diagramme UML

## 3 Commentaire

La class Main contient de nombreuses méthodes permettant notamment de comparer les combinaisons et trouver qu'elles est la plus forte.

## 4 main

```
1 package org.example;
```

```
2
3 import edu.info0502.poker.*;
4 import java.util.ArrayList;
5 import java.util.List;
6
7 import javax.print.attribute.standard.Media;
8
9 public class App {
10     public static void main(String[] args) {
11         // TP2
12         Carte c1 = new Carte(Couleur.Pique,
13                               Vrai_Couleur.Rouge, 12);
14         // System.out.println(c1.toString());
15         Paquet paquet = new Paquet();
16         System.out.println(paquet.toString());
17
18         Talon talon = new Talon(3);
19         talon.melanger();
20         System.out.println(talon.toString());
21
22         Main main1 = new Main(talon, 5);
23         System.out.println(main1.toString());
24         Main main2 = new Main(talon, 5);
25         System.out.println(main2.toString());
26
27         if (main1.estPlusForte(main2)) {
28             System.out.println("Main 1 plus forte\n");
29         } else {
30             System.out.println("Main 2 plus forte\n");
31         }
32
33         Main main3 = new Main();
34         Main main4 = new Main();
```

```
35     for (int i = 0; i < 5; i++) {
36         main1.distribuerCarte(i, talon);
37         main2.distribuerCarte(i, talon);
38         main3.distribuerCarte(i, talon);
39         main4.distribuerCarte(i, talon);
40     }
41     System.out.println(main1.toString());
42     System.out.println(main2.toString());
43     System.out.println(main3.toString());
44     System.out.println(main4.toString());
45
46     if (main1.estPlusForte(main2) &&
47         main1.estPlusForte(main3) &&
48         main1.estPlusForte(main4)) {
49         System.out.println("Victoire joueur 1\n");
50     } else if (main2.estPlusForte(main1) &&
51         main2.estPlusForte(main3) &&
52         main2.estPlusForte(main4)) {
53         System.out.println("Victoire joueur 2\n");
54     } else if (main3.estPlusForte(main1) &&
55         main3.estPlusForte(main2) &&
56         main3.estPlusForte(main4)) {
57         System.out.println("Victoire joueur 3\n");
58     } else
59         System.out.println("Victoire joueur 4\n");
60
61     //
62     #####
63
64     // texas hold'em
65
66     Main[] Joueurs = new Main[4];
67     for (int i = 0; i < 4; i++) {
68         Joueurs[i] = new Main(talon, 2);
69     }
```

```
62     for (int i = 0; i < 2; i++) {
63         Joueurs[0].distribuerCarte(i, talon);
64         Joueurs[1].distribuerCarte(i, talon);
65         Joueurs[2].distribuerCarte(i, talon);
66         Joueurs[3].distribuerCarte(i, talon);
67     }
68     Main Croupier = new Main(talon, 5);
69     Carte[] cartesCroupier = Croupier.getMain();
70     List<Carte[]> CombiFortes = new ArrayList<>();
71
72     // Pour chaque joueur
73     for (int i = 0; i < 4; i++) {
74         List<Carte[]> combinaisons =
75             Joueurs[i].getAllCombinations(cartesCroupier);
76         Carte[] meilleureCombinaison =
77             combinaisons.get(0); // On part du principe que
78             la premi re est la plus forte
79
80         // Comparer chaque combinaison pour trouver la
81         plus forte
82         for (int j = 1; j < combinaisons.size(); j++) {
83             Carte[] combinaisonActuelle =
84                 combinaisons.get(j);
85
86             // Cr er des objets 'Main' temporaires pour
87             utiliser 'estPlusForte'
88             Main mainTemp1 = new
89                 Main(meilleureCombinaison);
90             Main mainTemp2 = new Main(combinaisonActuelle);
91
92             // Si la combinaison actuelle est plus forte,
93             on met jour la meilleure
94             // combinaison
95             if (mainTemp2.estPlusForte(mainTemp1)) {
```

```
88         meilleureCombinaison = combinaisonActuelle;
89     }
90 }
91
92     // Ajouter la combinaison la plus forte de ce
93     joueur dans la liste CombiFortes
94     CombiFortes.add(meilleureCombinaison);
95 }
96
97 // Affichage des meilleures combinaisons pour chaque
98 joueur
99 for (int i = 0; i < CombiFortes.size(); i++) {
100     Main combinaison = new Main(CombiFortes.get(i));
101     System.out.println("Combinaison forte du joueur "
102         + (i + 1) + " : ");
103     System.out.println(combinaison.toString());
104     System.out.println();
105 }
106
107 int JV = 0;
108
109 for (int i = 1; i < CombiFortes.size(); i++) {
110     Main maintemp1 = new Main(CombiFortes.get(i));
111     Main maintemp2 = new Main(CombiFortes.get(JV));
112
113     if (maintemp1.estPlusForte(maintemp2)) {
114         JV = i;
115     }
116 }
117
118 Main JoueurGagnant = new Main(CombiFortes.get(JV));
119 System.out.println("Victoire joueur : " + (JV + 1) + "
120     avec : \n" + JoueurGagnant.toString());
121 }
```

## 5 Texas Hold'em

La partie de texas hold'em est implémenté principalement dans le main, en créant une nouvelle main Croupier et en donnant 2 cartes à chaque joueur. On cherche ensuite toutes les combinaisons de 5 cartes avec les 2 cartes du joueur et celle du croupier, puis on récupère la plus forte de toutes. On fait ça pour tout les joueurs et on compare enfin les combinaisons les plus fortes de tout les joueurs pour celui qui sera vainqueur.