Exercices PHP Objet 2

Génération de jeux de cartes

CONTENU

CONSIGNES

L'objectif de ces exercices est de générer divers jeux de cartes pour différents types de jeux et d'implémenter la distribution des cartes entre plusieurs joueurs selon les règles fixées par le type de jeu.

Créer un répertoire « php_objet_cartes » dédié aux exercices de ce document.

Dans cette série d'exercice, il vous sera demandé de :

- Implémenter des classes avec PHP.
- Créez des Tests liés à ces classes.

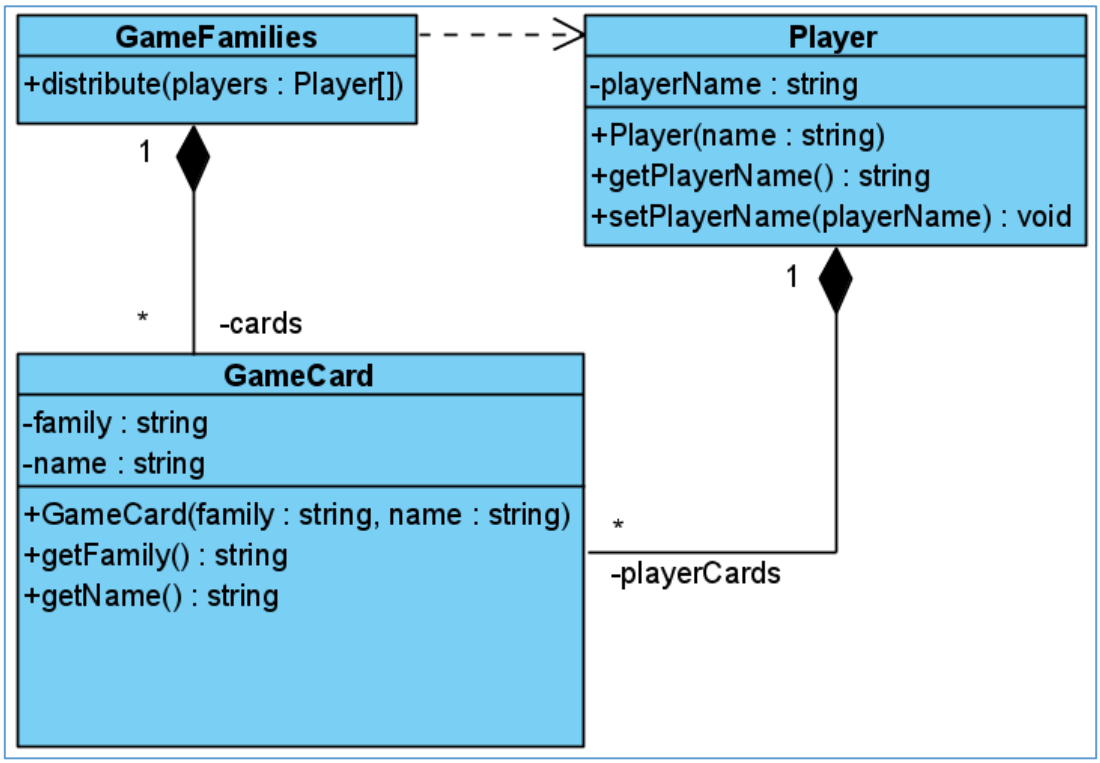
1 – 7 FAMILLES

Créez un répertoire « Familles ». Les fichiers des exercices suivants sont à placer dans ce répertoire.
Ce répertoire représente le « package » Families.
Les classes de ce répertoire font partie de l'espace de noms « Families » (namespace Families).

Le jeu des 7 familles est un jeu de cartes à plusieurs dont l'objectif est d'être le joueur ayant réussi à réunir le plus de familles complètes à la fin de la partie.

Le jeu est composé de 7 familles : Pierre, Paul, Jacques, Dupont, Martin, Stark et Lannister.
Chaque famille est composée de 6 membres : Le Grand-Père, La Grand-Mère, Le Père, La Mère, Le Fils et La Fille.

Le diagramme suivant représente les classes attendues.



Construisez le squelette de vos classes à partir de ce diagramme.

GÉNÉRER LE JEU DE CARTES

Le constructeur de la classe **GameFamilies** initialise le jeu de carte.

À partir de 2 tableaux :

```
$families = [ 'Pierre', 'Paul', 'Jacques', 'Dupont', 'Martin', 'Stark', 'Lannister' ];

$people = [ 'Le Grand-Père', 'La Grand-Mère', 'Le Père', 'La Mère', 'Le Fils', 'La Fille' ];
```

Générer le jeu des 7 familles dans le tableau « **cards** » de la classe **GameFamilies**. Chaque carte est représentée par une instance de la classe **GameCard**.

Les cartes doivent ensuite être mélangées.
Utilisez la fonction shuffle() de PHP pour mélanger le jeu de cartes.

DISTRIBUER LES CARTES

La méthode **distribute()** de la classe GameFamilies permet de distribuer les cartes entre plusieurs joueurs.

Cette méthode accepte un argument qui est un tableau de **Player**. Ce tableau doit contenir au minimum 2 et au maximum 4 **Player**.

Distribuez les cartes entre les joueurs en respectant les règles suivantes :

- On distribue à tour de rôle une carte par joueur.
- Chaque joueur doit posséder 7 cartes.

À la fin de la distribution, si un des joueurs possède les 6 membres d'une même famille, on recommence la distribution.

TESTER LES CLASSES

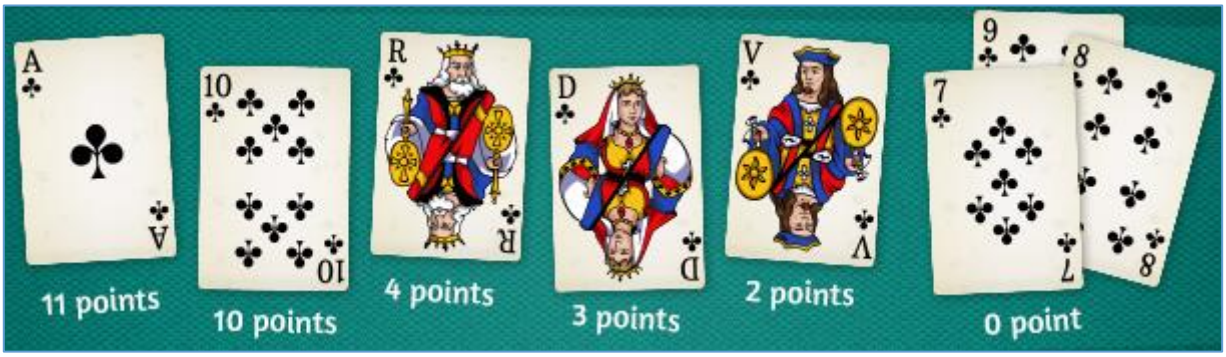
Dans un fichier **test_Families.php**, reprenez le code suivant pour tester vos classes :

```php
<?php
require 'GameCard.php' ;
require 'Player.php' ;
require 'GameFamilies.php' ;

use Families\GameCard ;
use Families\Player ;
use Families\GameFamilies ;

$players = [
0 => new Player('Mike'),
1 => new Player('Paul'),
2 => new Player('Cindy'),
] ;

$game = new GameFamilies() ; // génère le jeu de carte
var_dump($game) ;

$game->distribute($players) ; // distribue les cartes aux joueurs
var_dump($players) ;
```

2 – BELOTE

Créez un répertoire « Belote ». Les fichiers des exercices suivants sont à placer dans ce répertoire.
Ce répertoire représente le « package » Belote.
Les classes de ce répertoire font partie de l'espace de noms « Belote » (namespace Belote).

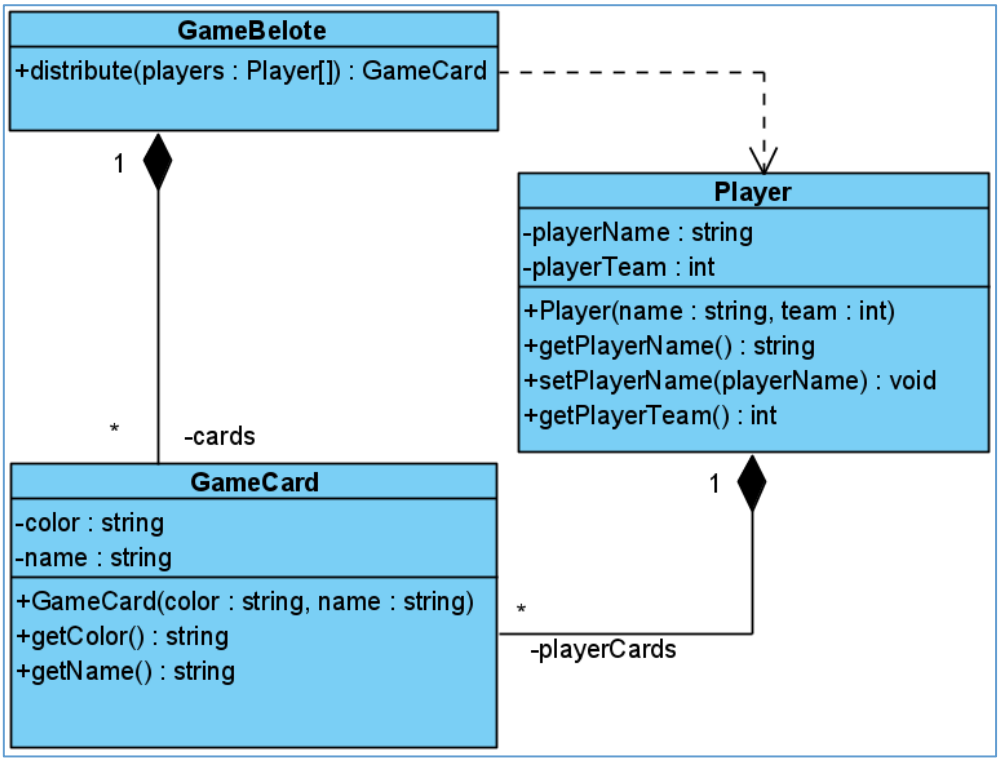
La Belote est un jeu de cartes populaire qui se joue à 4 joueurs (2 équipes de 2 joueurs).
L'objectif de la belote est d'obtenir un certain nombre de points.



Le jeu de cartes utilisé est composé de 4 couleurs : Pique, Cœur, Carreau et Trèfle.
Pour chaque couleur, il y a 8 cartes (par ordre de valeur) :

- Sept, Huit, Neuf, Dix, Valet, Dame, Roi, As

Le diagramme suivant représente les classes attendues.



Construisez le squelette de vos classes à partir de ce diagramme.

GÉNÉRER LE JEU DE CARTES

Le constructeur de la classe **GameBelote** initialise le jeu de carte.

À partir de 2 tableaux :

```
$colors = [ 'Pique', 'Coeur', 'Carreau', 'Trèfle' ];

$names = [ 'Sept', 'Huit', 'Neuf', 'Dix', 'Valet', 'Dame', 'Roi', 'As' ];
```

Générer le jeu de la Belote dans le tableau « **cards** » de la classe **GameBelote**. Chaque carte est représentée par une instance de la classe **GameCard**.

Les cartes doivent ensuite être mélangées.
Utilisez la fonction shuffle() de PHP pour mélanger le jeu de cartes.

DISTRIBUER LES CARTES

Dans le jeu de la Belote, les cartes sont distribuées en 2 temps :

1) 5 cartes par joueur
2) Choix de l'atout
3) Distribution du reste des cartes

La méthode **distribute()** de la classe **GameBelote** permet de distribuer les 5 premières cartes entre les 4 joueurs.

Cette méthode accepte un argument qui est un tableau de **Player**. Ce tableau doit contenir 4 **Player**.
Dans ce tableau, chaque joueur appartient à une équipe (1 ou 2).

- 2 joueurs dans l'équipe 1
- 2 joueurs dans l'équipe 2

Distribuez les cartes entre les joueurs en respectant les règles suivantes :

- On distribue 3 cartes par joueur, à tour de rôle.
  - 3 cartes au 1er joueur de l'équipe 1
  - 3 cartes au 1er joueur de l'équipe 2
  - 3 cartes au 2nd joueur de l'équipe 1
  - 3 cartes au 2nd joueur de l'équipe 2
- On distribue ensuite 2 cartes supplémentaires par joueur, à tour de rôle dans le même ordre.

À cette étape, chaque joueur doit posséder 5 cartes.

Une fois ces 20 cartes distribuées, la carte suivante dans le paquet est retournée et affichée aux joueurs.
Les joueurs doivent ensuite choisir un « Atout ».

À la fin de la distribution, La méthode **distribute()** doit **retourner** une instance de GameCard correspondant à la carte suivante dans le paquet de cartes (parmi celles non distribuées).

TESTER LES CLASSES

Inspirez-vous du code d'exemple de l'exercice précédent pour effectuer vos tests.

3 – FUSION (BONUS)

Les 2 exercices précédents présentent des similitudes au niveau de l'architecture et donc du code.
Certaines classes des 2 exercices sont relativement proches dans leur structure.
Peut-être aurez-vous remarqué des portions de code identiques dans les 2 exercices.

Vous avez maintenant le devoir de créer des classes communes aux 2 jeux pour éliminer la redondance de code.

Proposez une solution qui permettrait de rassembler les similitudes des 2 jeux.

Pour cela, utilisez **l'héritage**, les **classes abstraites** et les **interfaces**.

--- FIN DU DOCUMENT ---