

La Programmation C++

Exercices Partie 1

1 - Ecrire un programme C++ qui demande à l'utilisateur

1.1 - de saisir son nom et de lui afficher son nom avec le message de bienvenue ensuite

1.2 - de saisir successivement deux nombres réels et de lui afficher le maximum des deux ensuite

1.3 - de saisir un nombre entier et de lui afficher que le nombre est pair ou impair selon la valeur tapée

Ecrire un programme C++ qui calcul

- la somme des 20 premiers entiers
- le factoriel de 20

TP 1.3 – binaire

Ecrire un programme permettant de :

- Transformer un nombre binaire vers son équivalent décimale
- Transformer un nombre décimal en binaire

Exemple de résultat demandé :

Voulez vous convertir :

1- un nombre décimal en binaire

2- un nombre binaire en décimal

Rentrer votre choix : 2

Rentrer un nombre binaire : 11

Résultat : 11 vos 3 en décimal

Écrire un jeu dont le but est de trouver un nombre choisi par la machine compris entre -N et N

Exemple d'exécution :

```
> NombreMystere 10
```

```
Le nombre à trouver est compris entre -10 et 10
```

```
Entrer un nombre :
```

```
0
```

```
Trop petit
```

```
5
```

```
Trop grand
```

```
1
```

```
Bravo le chiffre est bien 1, vous avez gagné en 3 coups .
```

```
Voulez vous refaire une partie o[O] ? N
```

```
>
```

Faire un programme qui simule le lance de 3 dés
Et qui permet d'obtenir en 3 coups max : 4 2 1

Exemple : joue 3 des : 5 2 1 garde 2 et 1 [2,1]
 joue 1 des : 3 garde rien [2,1]
 joue 1 des :4 garde 4 [4,2,1]
 gagné

TP 1.6 - Jeu des allumettes

La règle : il y a plusieurs allumettes(autant qu'on le veut) et on en retire 1,2 ou 3 et celui qui prend la dernière a perdu.

Exemple :

Choisir le nombre d'allumette de départ : 6

||||| joueur 1 enlève : 1

||||| joueur 2 enlève : 2

||| joueur 1 enlève : 2

| joueur 2 enlève : 1

le joueur 2 a perdu :-(

Faire un programme qui calcule l'intégrale de la fonction $y = x^2$ par la méthode des rectangles avec $x \in [a,b]$ et un pas p

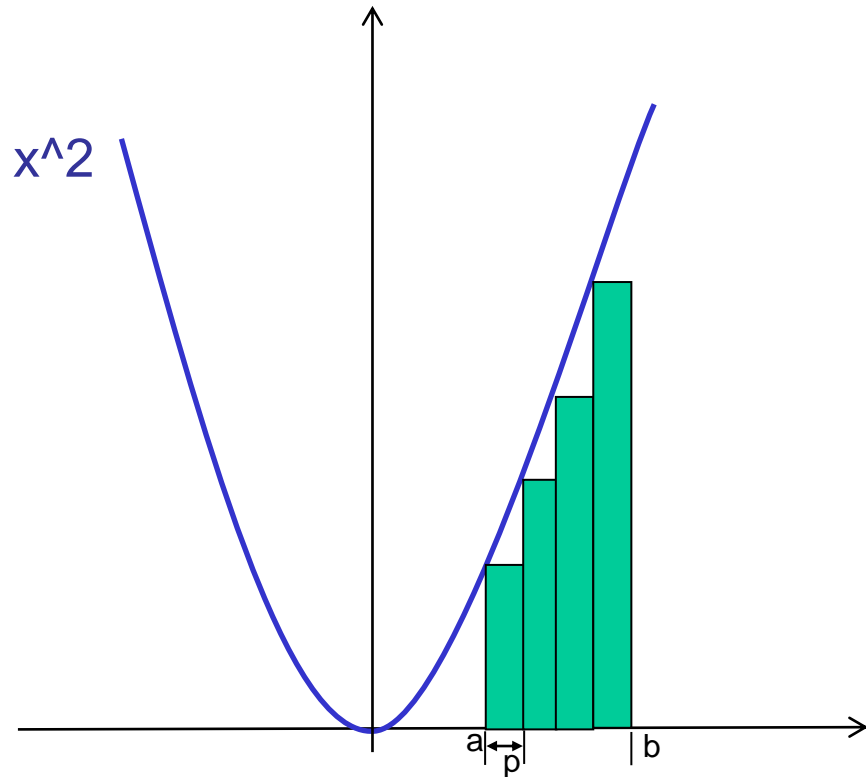
Exemple d'exécution :

> Integrale 3 5 5

Calcul de l'intégrale de la fonction $y = x^2$
avec $3 < x < 5$ et $p = 0.4$

Résultat 32.72

>



Écrire une fonction maxfact qui pour un entier k donné en paramètre calcule le plus grand entier n tel que $n! \leq k$.

Exemple:

```
$> maxfact 121
```

```
5! <= 121
```

```
$>
```

La Programmation C++

Exercices Partie 2

TP 2.1a - Référence/Pointeur : Corriger et Compléter

```
#include <iostream>
```

```
typedef double* ptrDouble;
```

```
void constructeur(ptrDouble dd,const unsigned & taille) { ACOMPLETER }
```

```
void afficher (const ptrDouble dd, const unsigned& taille) { ACOMPLETER }
```

```
void modifier (ptrDouble const dd, const unsigned& taille, const unsigned& index, const double& valeur) { ACOMPLETER }
```

```
void destructeur (ptrDouble dd) { ACOMPLETER }
```

```
const double& get(const ptrDouble dd, const unsigned& taille, const unsigned& index { ACOMPLETER }
```

```
double& get(ptrDouble dd, const unsigned& taille, const unsigned& index) { ACOMPLETER }
```

```
void C_2_6b() {
```

```
    ptrDouble d1 = nullptr;
```

```
    unsigned t1=5;
```

```
    constructeur(d1,t1);
```

```
    afficher(d1,t1);
```

```
    modifier(d1,t1,2, 3.13589985);
```

```
    afficher(d1,t1);
```

```
    std::cout<< get(d1,t1,2) << std:: endl;
```

```
    get(d1,t1,2) = 62.1;
```

```
    std::cout<< get(d1,t1,2) << std:: endl;
```

```
    afficher(d1,t1);
```

```
    destructeur(d1);
```

```
    afficher(d1,t1);
```

```
}
```

TP 2.1b - Référence/Pointeur intelligent Référence/Pointeur : Corriger et Compléter

```
#include <iostream>
#include <memory>
```

```
typedef std::shared_ptr<double> ptrStdDouble;
```

```
void constructeur(ptrStdDouble dd,const unsigned & taille) { ACOMPLETER }
void destructeur (ptrStdDouble dd) { ACOMPLETER }
void afficher (const ptrStdDouble dd, const unsigned& taille) { ACOMPLETER }
void modifier (ptrStdDouble const dd, const unsigned& taille, const unsigned& index, const double& valeur) { ACOMPLETER }
const double& get(const ptrStdDouble dd, const unsigned& taille, const unsigned& index) { ACOMPLETER }
double& get( ptrStdDouble dd, const unsigned& taille, const unsigned& index) { ACOMPLETER }
```

```
void C_2_6c() {
    ptrStdDouble d1 = nullptr;
    unsigned t1=5;
    constructeur(d1,t1);
    afficher(d1,t1);
    modifier(d1,t1,2, 3.13589985);
    afficher(d1,t1);
    std::cout<< get(d1,t1,2) << std:: endl;
    get(d1,t1,2) = 62.1;
    std::cout<< get(d1,t1,2) << std:: endl;
    afficher(d1,t1);
    destructeur(d1); }
```

- Ecrire une macro de debuggage qui affiche systématiquement le numéro de la ligne et le nom du fichier
- Ecrire une macro qui rend le maximum de deux valeurs

Faire un programme qui calcule une suite de Syracuse telle que

La suite de Syracuse d'un nombre entier $N > 0$ est **définie par récurrence**, de la manière suivante :

$$u_0 = N$$

$$\text{et pour tout entier naturel } n : u_{n+1} = \begin{cases} \frac{u_n}{2} & \text{si } u_n \text{ est pair,} \\ 3u_n + 1 & \text{si } u_n \text{ est impair.} \end{cases}$$

Énoncé de la conjecture [\[modifier \]](#) [\[modifier le code \]](#)

La **conjecture** affirme que pour tout N , il existe un indice n tel que $u_n = 1$.

Suite de Syracuse pour $N = 15$

u_0	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}	u_{11}	u_{12}	u_{13}	u_{14}	u_{15}	u_{16}	u_{17}	u_{18}	u_{19}	u_{20}	
15	46	23	70	35	106	53	160	80	40	20	10	5	16	8	4	2	1	4	2	1	...

https://fr.wikipedia.org/wiki/Conjecture_de_Syracuse

TP 2.4 - Fibonacci

Écrire une fonction calculant le nombre de Fibonacci d'un nombre passé en paramètre.
Le nombre de Fibonacci $F(n)$ est défini comme suit :

$$\begin{aligned} F(0) &= 1; \\ F(1) &= 1; \\ F(n) &= F(n-1) + F(n-2) \end{aligned}$$

En mathématiques, la **suite de Fibonacci** est une suite d'entiers dans laquelle chaque terme est la somme des deux termes qui le précèdent. Elle commence par les termes 0 et 1 (on trouve des définitions [\[réf. nécessaire\]](#) qui la font commencer avec 1 et 1). Les termes de cette suite sont appelés *nombre de Fibonacci* (suite [A000045](#) de l'OEIS) :

\mathcal{F}_0	\mathcal{F}_1	\mathcal{F}_2	\mathcal{F}_3	\mathcal{F}_4	\mathcal{F}_5	\mathcal{F}_6	\mathcal{F}_7	\mathcal{F}_8	\mathcal{F}_9	\mathcal{F}_{10}	\mathcal{F}_{11}	\mathcal{F}_{12}	\mathcal{F}_{13}	\mathcal{F}_{14}	\mathcal{F}_{15}	\mathcal{F}_{16}	...	\mathcal{F}_n
0	1	1	2	3	5	8	13	21	34	55	89	144	233	377	610	987	...	$\mathcal{F}_{n-1} + \mathcal{F}_{n-2}$

La suite est définie par $\mathcal{F}_0 = 0$, $\mathcal{F}_1 = 1$, et $\mathcal{F}_n = \mathcal{F}_{n-1} + \mathcal{F}_{n-2}$, pour $n > 1$.

https://fr.wikipedia.org/wiki/Suite_de_Fibonacci

```
enum Sexe { INCONNUE=0,MASCULIN=1,FEMININ=2};  
  
struct Personne {  
    int numero;  
    char nom[10];  
    Sexe sexe;  
};
```

Écrire les fonctions nommées suivantes :

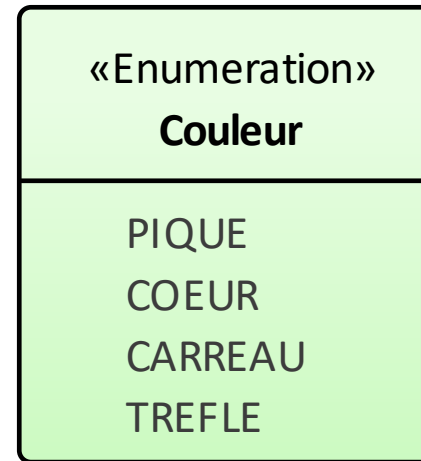
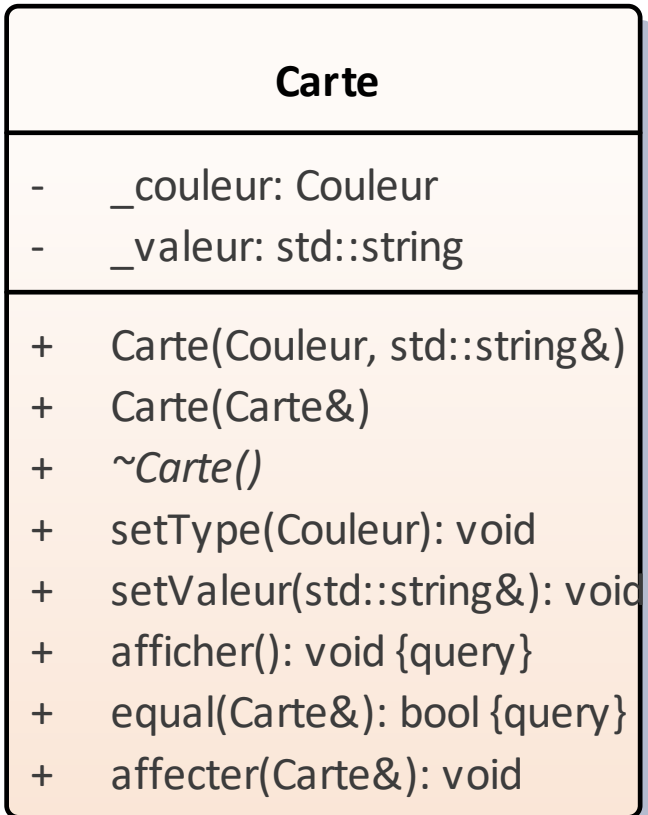
- **créer** permettant de créer un pointeur de la structure Personne
- **détruire** permettant de détruire un pointeur de la structure Personne
- **initialiser** permettant d'initialiser les champs de la structure Personne
- **afficher** permettant d'afficher les champs de la structure Personne

La Programmation C++

Exercices Partie 3

TP 3.1 – Jeu de carte (1)

class Domain Model



TP 3.1 – Jeu de carte (1)

```
#include <iostream>
#include "Carte.h"

using namespace std;

//enum Couleur{ PIQUE,COEUR,CARREAU, TREFLE};

int main()
{
    cout << "Jeu de carte" << endl;
    Carte c1(PIQUE, "As");
    c1.afficher();
    Carte c2 (c1);
    c2.afficher();
    c2.setType(TREFLE);
    c2.setValeur("Queen");
    c2.afficher();
    Carte c3(PIQUE, "2");
    c2.affecter(c3);
    c2.afficher();
    c3.afficher();

    if ( c1.equal(c2) ) {
        cout << "is ok :-)" << endl;
    } else {
        cerr << " problem bug" << endl;
        c1.afficher();
        c2.afficher();
    }

    return 0;
}
```

TP 3.2 – Jeu de carte (2)

class Domain Model

Carte

```

+  NbCreation: unsigned
-  _couleur: Couleur
-  _valeur: std::string

+  Carte(Couleur, std::string&)
+  Carte(Carte&)
+  ~Carte()
+  operator=(Carte&): Carte&
+  operator==(Carte&): bool {query}
+  operator!=(Carte&): bool {query}
+  setType(Couleur): void
+  setValeur(std::string&): void

«property get»
+  GetNbCreation(): unsigned

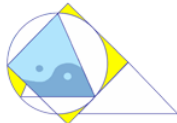
«friend»
+  operator<<(std::ostream&, Carte&): std::ostream&
    
```

class Domain Model

«Enumeration» Couleur

```

PIQUE
COEUR
CARREAU
TREFLE
    
```



TP 3.2 – Jeu de carte (2)

```
#include <iostream>
#include "Carte.h"

using namespace std;

int main()
{
    cout << "Jeu de carte" << endl;
    Carte c1(PIQUE, "As");
    cout << c1 << endl;

    Carte c2 (c1);
    cout << c2 << endl;
    c2.setType(TREFLE);
    c2.setValeur("Queen");
    cout << c2 << endl;

    if ( c1 != c2 ) {
        cout << "is ok :-)" << endl;
    } else {
        cout << " problem bug" << endl;
    }
    return 0;
}
```

Règle du jeu

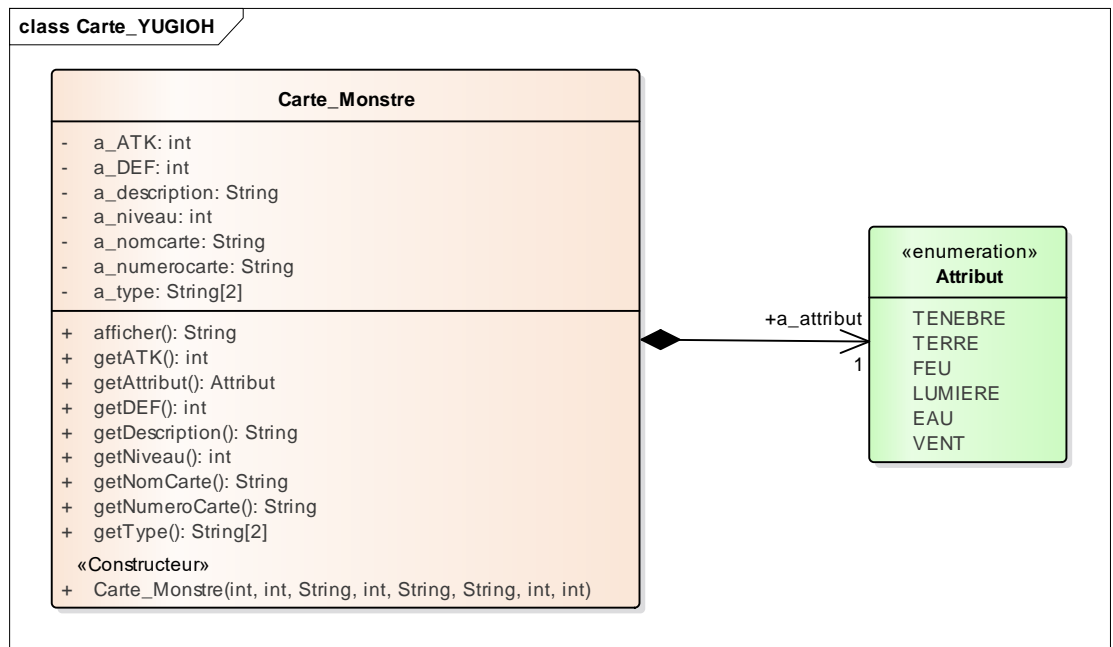
https://img.yugioh-card.com/ygo_cms/ygo/all/uploads/Rulebook_v9_fr.pdf
<https://www.kingyugi.fr/les-regles-et-bases-du-yu-gi-oh/>

Toutes les Cartes Yu-Gi-Oh!

www.finalyugi.com/yugioh-cartes.html



Code la carte Monstre et vérifier que les méthodes fonctionnent.



Remarques : le diagramme donnée n'est pas exhaustive et peut être compléter et améliorer

TP 3.4 – Jeu de carte YOGIOH

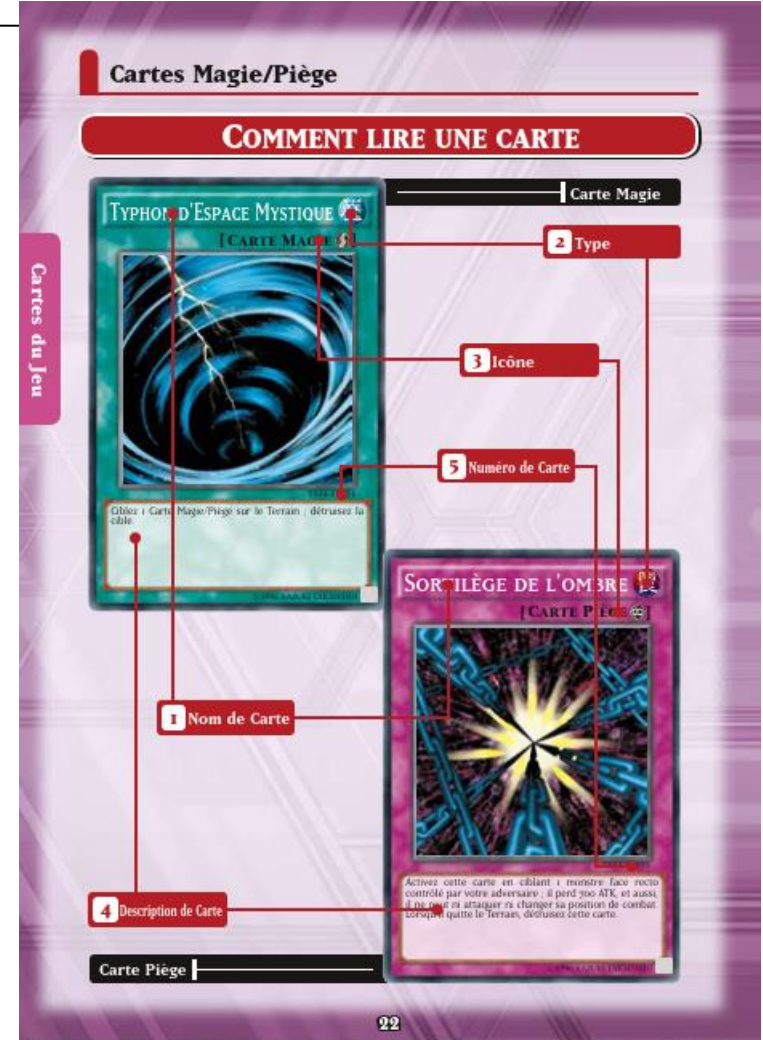
Règle du jeu

https://img.yugioh-card.com/ygo_cms/ygo/all/uploads/Rulebook_v9_fr.pdf

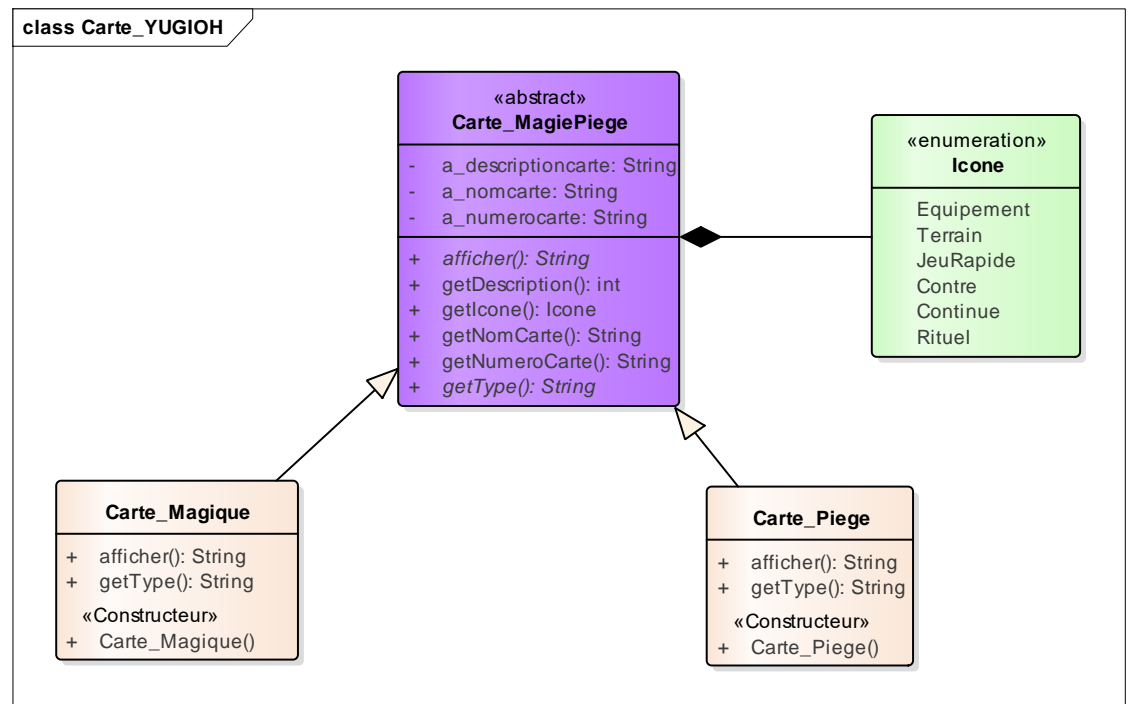
<https://www.kingyugi.fr/les-regles-et-bases-du-yu-gi-oh/>

Toutes les Cartes Yu-Gi-Oh!

www.finalyugi.com/yugioh-cartes.html



Coder les cartes Magie et Piège et vérifier que les méthodes fonctionnent.



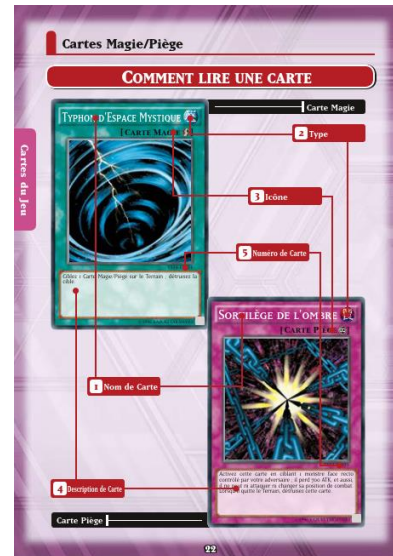
Remarques : le diagramme donnée n'est pas exhaustive et peut être compléter et améliorer

Règle du jeu

https://img.yugioh-card.com/ygo_cms/ygo/all/uploads/Rulebook_v9_fr.pdf

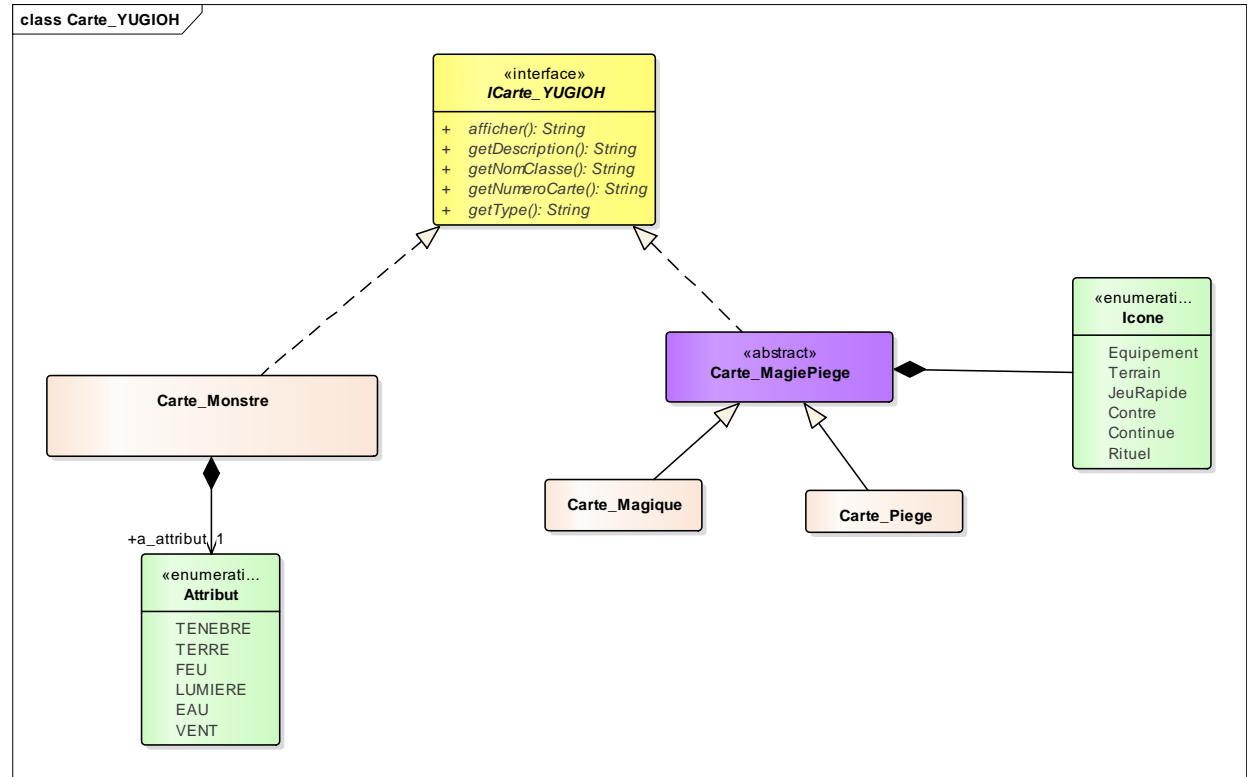
Toutes les Cartes Yu-Gi-Oh!

www.finalyugi.com/yugioh-cartes.html



Généralisez la notion de carte YUGIOH

Remarques : le diagramme donné n'est pas exhaustive et peut être compléter et améliorer



TP 3.6 – Jeu de carte YOGIOH (bonus)

- Créer une classe *FabriqueCarte_YUGIOH* qui créer des cartes YUGIOH d'un certain type au hasard et qui les stocke dans une classe *JeuDeCarteYOGIOH* .

```
class Carte_YUGIOH
```

```
    «Fabrique»  
    FabriqueCarte_YUGIOH
```

```
+ CreerCarte(String): ICarte_YUGIOH
```



Remarques : le diagramme donnée n'est pas exhaustive et peut être complété et amélioré

[https://fr.wikipedia.org/wiki/Fabrique_\(patron_de_conception\)](https://fr.wikipedia.org/wiki/Fabrique_(patron_de_conception))
<https://gfox.dev/developpez-votre-tutoriel/conception/pattern/fabrique/>

