# La Programmation C++

# Exercices Partie 1


Copyright : Yantra Technologies 2004-2019

# TP 1.1 - Nombre mystère

**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
>
```


Copyright : Yantra Technologies 2004-2019

# TP 1.2 - 421

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


Copyright : Yantra Technologies 2004-2019

# TP 1.3 - 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 :-(


Copyright : Yantra Technologies 2004-2019

# TP 1.4 - Syracuse

## 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 :

$$u_0 = N$$

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

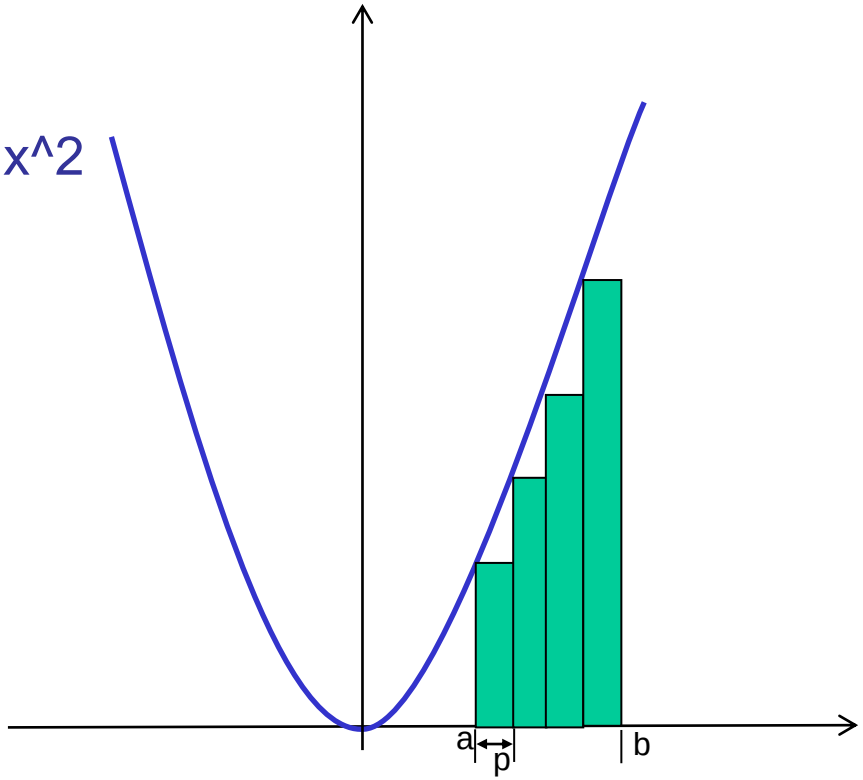

Copyright : Yantra Technologies 2004-2019

# TP 1.5 - Intégrale

**Faire un programme qui calcule l'intégrale de la fonction y = x^2 par la méthode des rectangles avec x∈ [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
>
```





Copyright : Yantra Technologies 2004-2019

# TP 1.6 - 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 :**

**F(0) = 1;**
**F(1) = 1;**
**F(n) = F(n - 1) + F(n-2)**

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 *nombres 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 $\mathcal{F}_0 = 0, \quad \mathcal{F}_1 = 1$, et $\mathcal{F}_n = \mathcal{F}_{n-1} + \mathcal{F}_{n-2}$, pour $n > 1$.

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


Copyright : Yantra Technologies 2004-2019

# TP 1.7 – Développements limités

- Coder 3 développements limités usuels,
- Et vérifier le résultat

## Développements limités usuels

$$e^x = 1 + x + \frac{x^2}{2} + \cdots + \frac{x^n}{n!} + o(x^n)$$

$$\cos x = 1 - \frac{x^2}{2!} + \cdots + \frac{(-1)^n x^{2n}}{(2n)!} + o(x^{2n+1})$$

$$\sin x = x - \frac{x^3}{3!} + \cdots + \frac{(-1)^n x^{2n+1}}{(2n+1)!} + o(x^{2n+2})$$

$$\cosh x = 1 + \frac{x^2}{2!} + \cdots + \frac{x^{2n}}{(2n)!} + o(x^{2n+1})$$

$$\sinh x = x + \frac{x^3}{3!} + \cdots + \frac{x^{2n+1}}{(2n+1)!} + o(x^{2n+2})$$

$$\frac{1}{1-x} = 1 + x + x^2 + \cdots + x^n + o(x^n)$$

$$\ln(1+x) = x - \frac{x^2}{2} + \cdots + \frac{(-1)^{n+1}}{n} x^n + o(x^n)$$

$$\arctan(x) = x - \frac{x^3}{3} + \cdots + \frac{(-1)^n}{2n+1} x^{2n+1} + o(x^{2n+1})$$

$$(1+x)^\alpha = 1 + \alpha x + \frac{\alpha(\alpha-1)}{2} x^2 + \cdots + \frac{\alpha(\alpha-1)\cdots(\alpha-n+1)}{n!} x^n + o(x^n)$$

$$\tan(x) = x + \frac{x^3}{3} + \frac{2}{15} x^5 + o(x^5).$$


Copyright : Yantra Technologies 2004-2019

# TP 1.8 : maxfact

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

Exemple:

$> maxfact 121

5! <= 121

$>



Copyright : Yantra Technologies 2004-2019

# La Programmation C++

# Exercices Partie 2


Copyright : Yantra Technologies 2004-2019

# TP 2.1a - Référence/Pointeur

Écrire un programme qui :

- Déclare un entier
- Déclare une référence vers cet entier
- Déclare un pointeur vers cet entier
- Dans les deux cas, imprimer la variable, l'adresse de la variable, la valeur pointée.

Écrire une fonction « affiche » qui affiche un pointeur, son adresse et sa valeur.

Ecrire une fonction « constructeur » qui prend un pointeur en argument et lui affecte une valeur.

Ecrire une fonction « destructeur » qui prend un pointeur en argument et libère la mémoire.



Copyright : Yantra Technologies 2004-2019

# TP 2.1b - 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);
}
```


Copyright : Yantra Technologies 2004-2019

# TP 2.1c - 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); }
```


Copyright : Yantra Technologies 2004-2019

# TP 2.2 - Macro

- 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


Copyright : Yantra Technologies 2004-2019

# TP 2.3 - Personne

```
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
- ***detruire*** 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

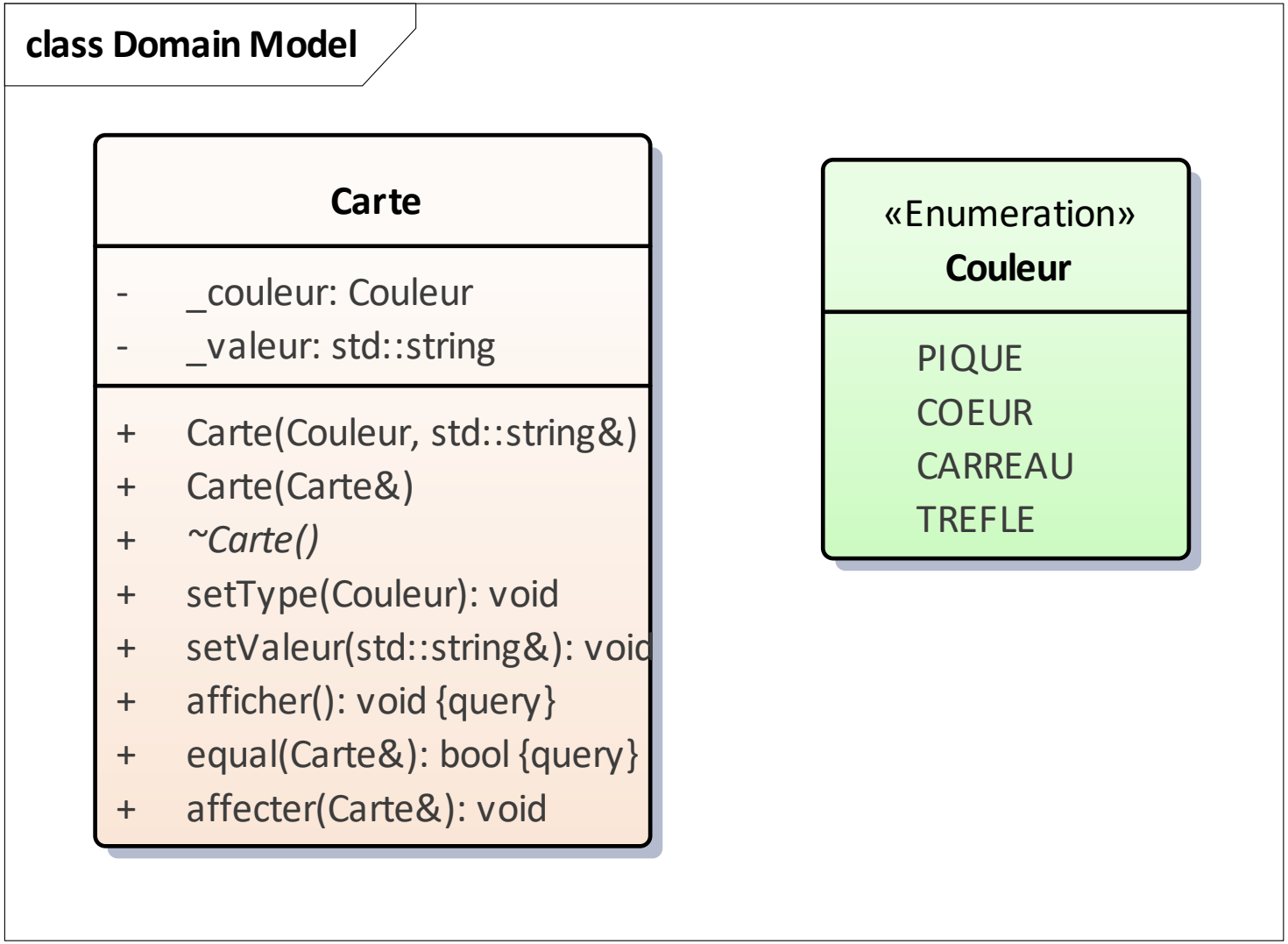

Copyright : Yantra Technologies 2004-2019

# La Programmation C++

# Exercices Partie 3


Copyright : Yantra Technologies 2004-2019

# TP 3.1 – Jeu de carte (1)





Copyright : Yantra Technologies 2004-2019

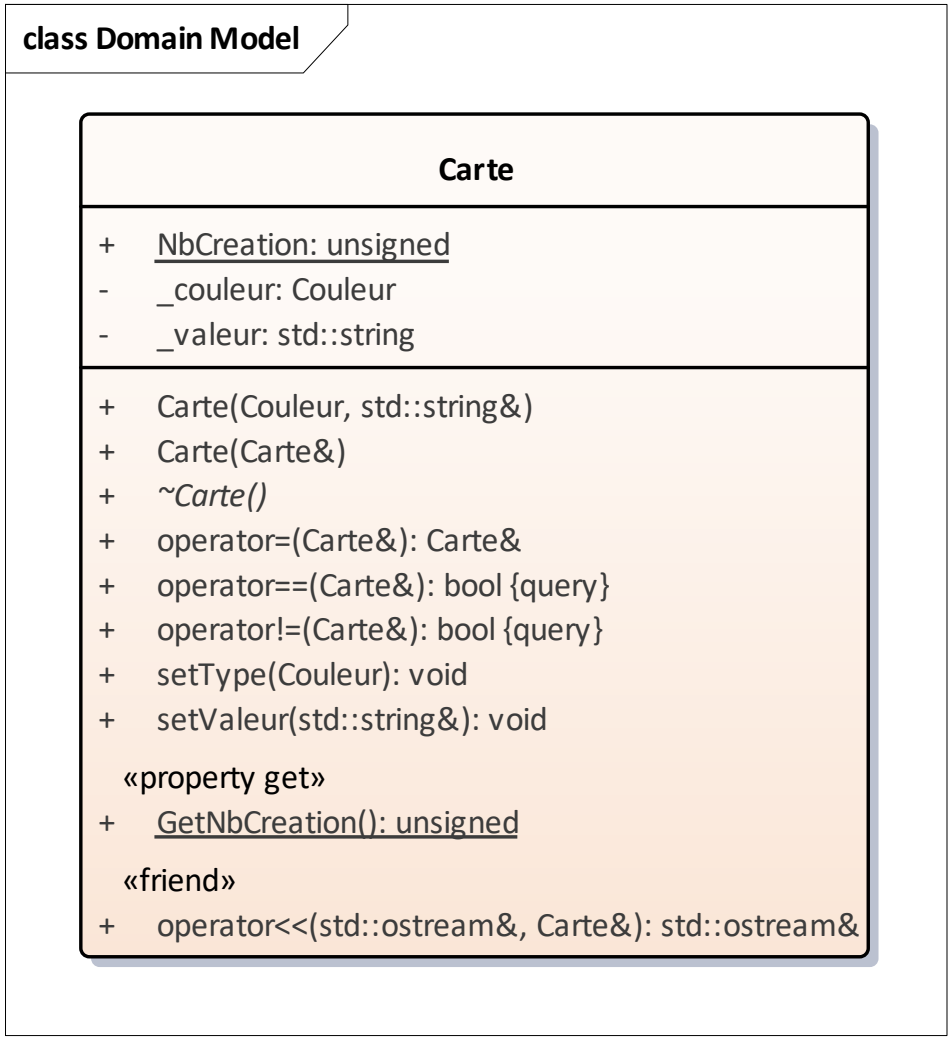
# TP 3.1 – Jeu de carte (1)

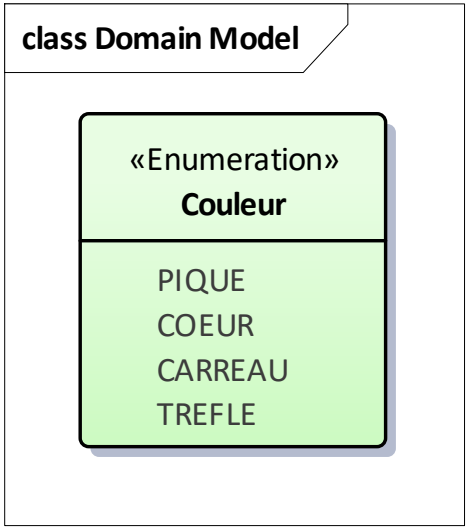
```
#include <iostream>
#include "Carte.h"

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;
}
```



Copyright : Yantra Technologies 2004-2019

# TP 3.2 – Jeu de carte (2)





Copyright : Yantra Technologies 2004-2019

# 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;
}
```


Copyright : Yantra Technologies 2004-2019

# TP 3.3 – 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!

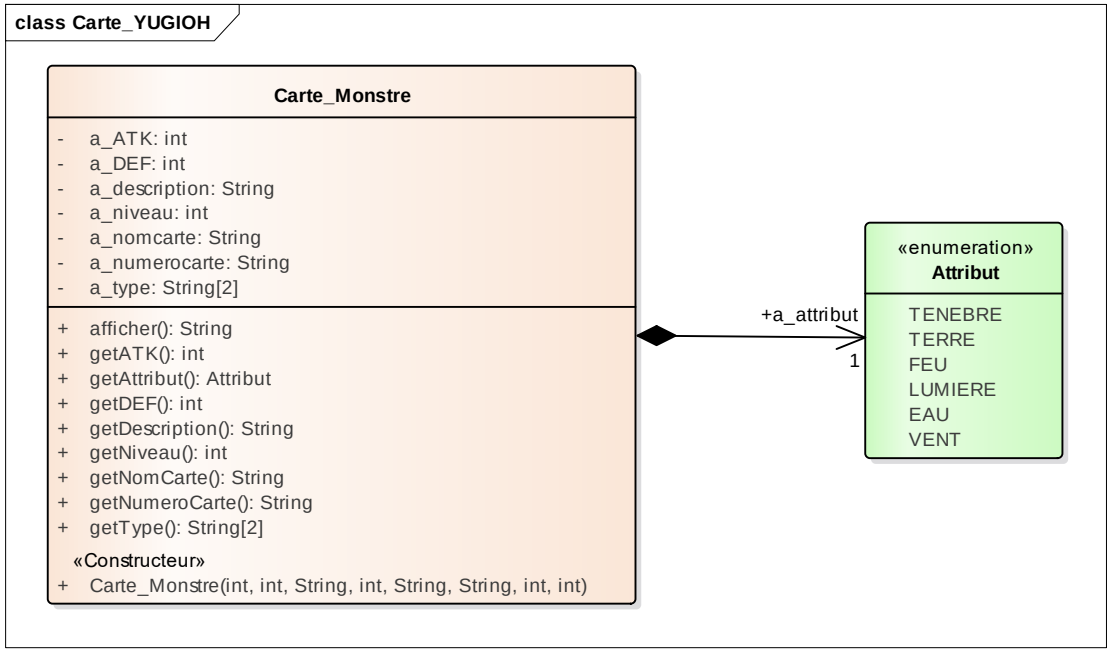
www.finalyugi.com/yugioh-cartes.html



Copyright : Yantra Technologies 2004-2019

# TP 3.3 – Jeu de carte YOGIOH

Code la carte Monstre et vérifier que les méthodes fonctionnent.




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


Copyright : Yantra Technologies 2004-2019

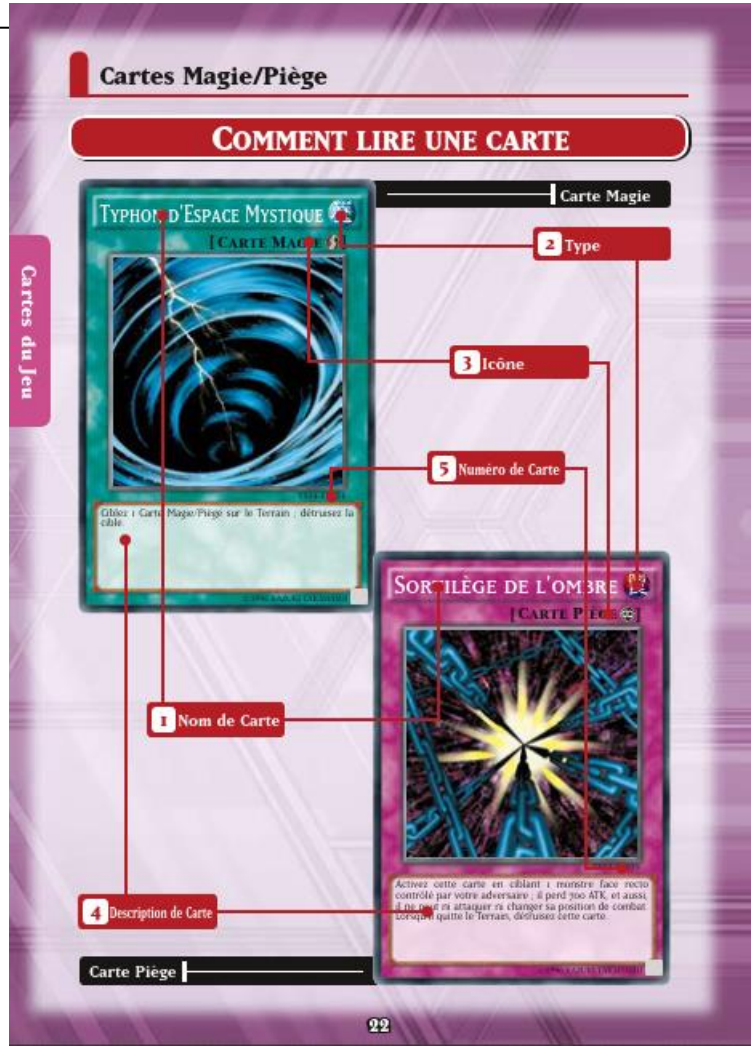
# TP 3.4 – Jeu de carte YOGIOH

**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



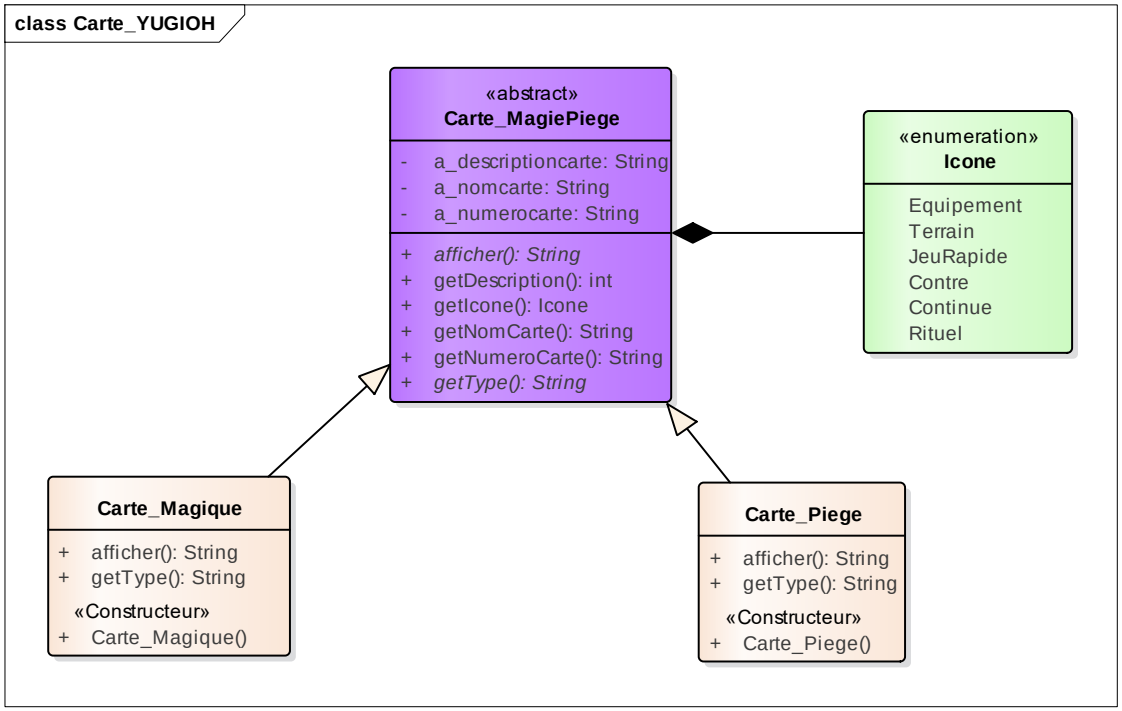


Copyright : Yantra Technologies 2004-2019

# TP 3.4 – Jeu de carte YOGIOH

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**





Copyright : Yantra Technologies 2004-2019

# TP 3.5 – Jeu de carte YOGIOH

**Règle du jeu**

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

**Toutes les Cartes Yu-Gi-Oh!**

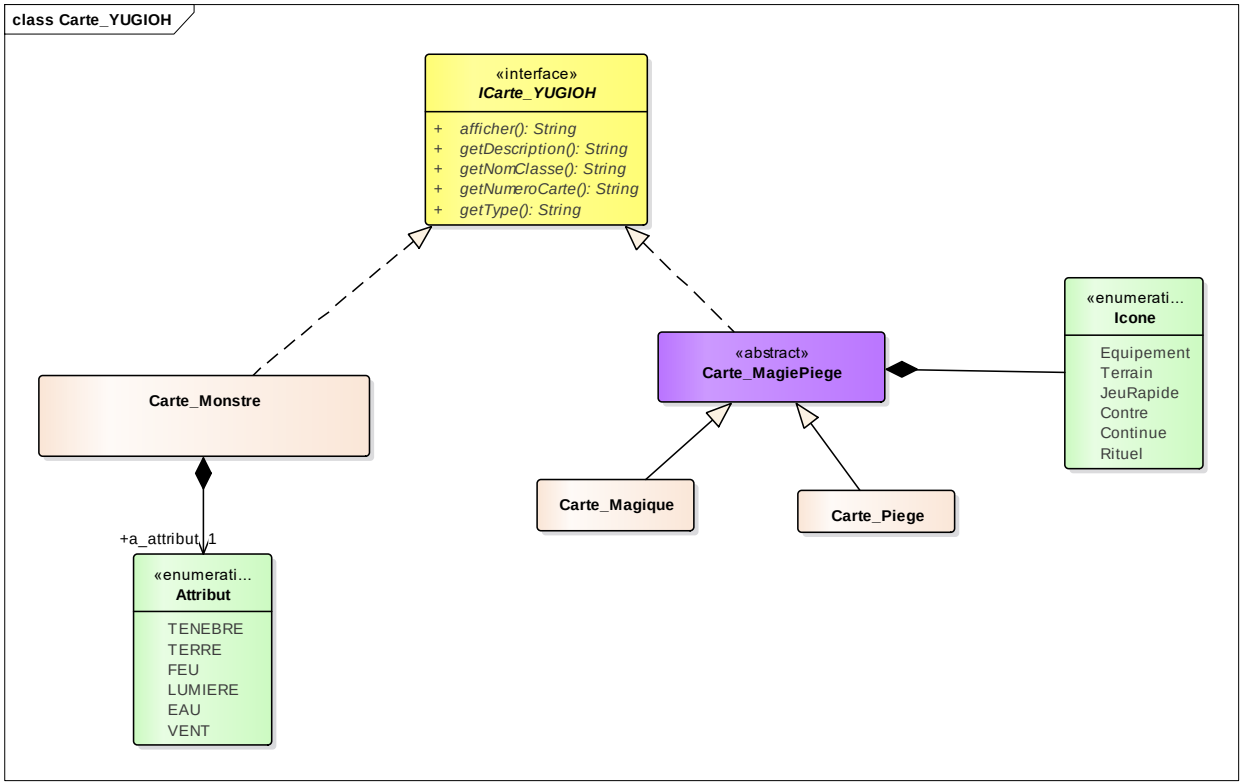
www.finalyugi.com/yugioh-cartes.html




Copyright : Yantra Technologies 2004-2019

# TP 3.5 – Jeu de carte YOGIOH

Généralisez la notion de carte YUGIOH

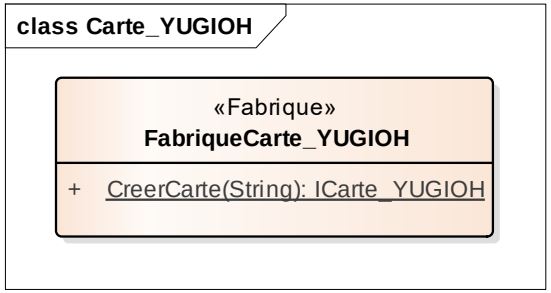



**Remarques : le diagramme donnée n'est pas exhaustive et peut être compléter et améliorer**


Copyright : Yantra Technologies 2004-2019

# TP 3.6 – Jeu de carte YOGIOH

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




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

https://fr.wikipedia.org/wiki/Fabrique_(patron_de_conception)
https://gfx.developpez.com/tutoriel/conception/pattern/fabrique/



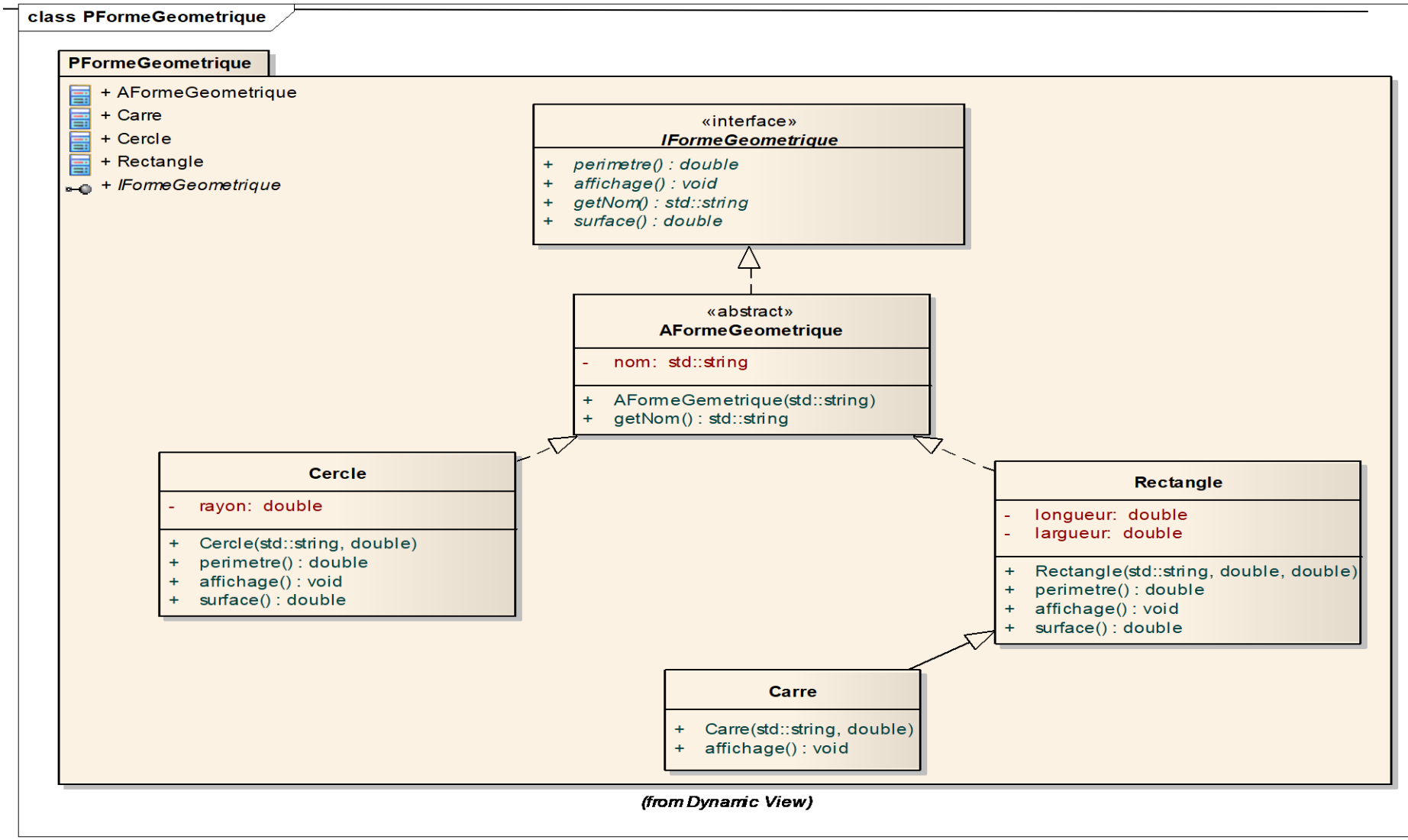
Copyright : Yantra Technologies 2004-2019

# La Programmation C++

# Exercices Partie 4


Copyright : Yantra Technologies 2004-2019

# TP 4.1 - Package PFormeGeometrique





Copyright : Yantra Technologies 2004-2019

# TP 4.1- Package PFormeGeometrique

Exemple :

-> makefile

-> inc

-> PFormeGeometrique :

-> IFormeGeometrique.hh

-> AFormeGeometrique.hh

-> Cercle.hh

-> Rectangle.hh

-> Carre.hh

-> src :

-> AFormeGeometrique.cpp

-> Cercle.cpp

-> Rectangle.cpp

-> Carre.cpp

-> main.cpp

-> obj :

-> AFormeGeometrique.o

-> Cercle.o

-> Rectangle.o

-> Carre.o

-> main.o

-> exe :

-> FormeGeometrique.exe



Copyright : Yantra Technologies 2004-2019

# TP 4.1 - Package PFormeGeometrique

```
#ifdef ISDEBUG
#include <iostream>
#define DEBUG(msg) { \
std::cerr<< « Debug(« <<__FILE__ <<», »<<__LINE__ <<») » << msg << std::endl; \
std::cerr.flush(); }
#else
#define DEBUG(msg)
#endif
```

```
namespace PFormeGeometrique {
	class IFormeGeometrique {
	public:
		virtual double perimetre() const = 0;
		virtual void affichage() const = 0;
		virtual std::string getNom() const = 0;
		virtual double surface() const = 0;

		virtual ~IFormeGeometrique() { DEBUG(« ~IFormeGeometrique() »); }
	};
}
```


Copyright : Yantra Technologies 2004-2019

# TP 4.1 - Package PFormeGeometrique

```
#include "Cercle.hh"
#include "Carre.hh"

using namespace PFormeGeometrique;

int main(int argc, char *argv[])
{
  Cercle c("C10",10.);
  c.affichage();
  c.perimetre();
  c.surface();

  Rectangle r("R10",10.,5.0);
  r.affichage();
  r.perimetre();
  r.surface();

  Carre ca("CA10",10.);
  ca.affichage();
  ca.perimetre();
  ca.surface();

  IFormeGeometrique *f = new Carre("CARR002",12);
  f->affichage();
  delete f;f=NULL;

  IFormeGeometrique &f2 = ca;
  f2.affichage();

  return 0;
}
```



Copyright : Yantra Technologies 2004-2019

# TP 4.2 - Package PFormeGeometrique

- Créer une fabrique ( voir design patterns) de forme géométrique


Copyright : Yantra Technologies 2004-2019