

# PROGRAMMATION ORIENTEE OBJET

## 1 – CORBEILLE

03/11/2020

---

### EXERCICES DE LA CORBEILLE

---

#### Exercice 1

1. Créez une classe 'CLa'
2. Cette classe doit comporter une méthode publique 'void afficher(void)
3. La méthode doit afficher 'bonjour' sur la sortie standard
4. Dans le main créez un objet o1 et un objet dynamique o2 ;
5. Allouez de la mémoire pour o2
6. Appelez afficher de o1
7. Appelez afficher de o2
8. Affichez l'adresse de o1
9. Affichez l'adresse de o2
10. Affichez l'adresse pointée par de o2
11. Affichez la taille en octets du type CLa
12. Affichez la taille en octets de o1
13. Affichez la taille en octets de o2
14. Libérez la mémoire pour o2

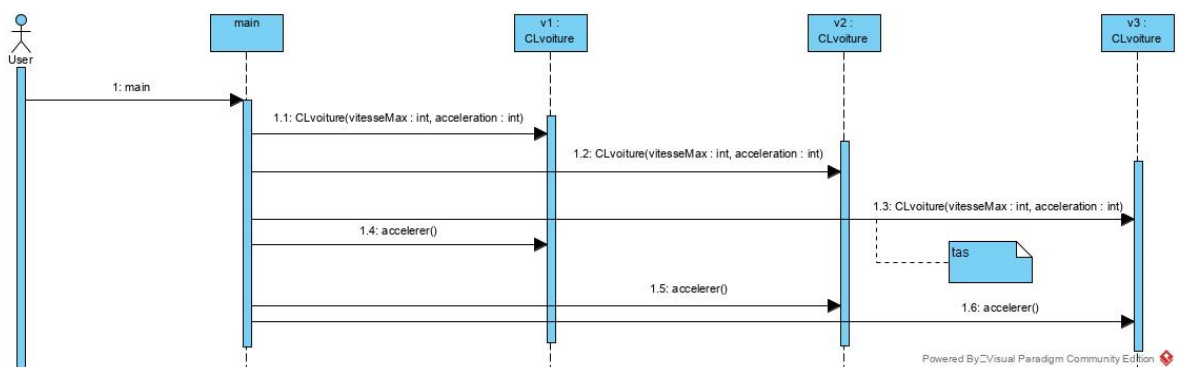
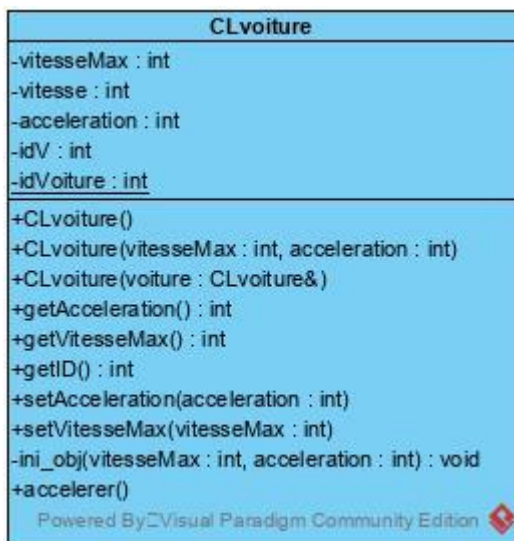
#### Exercice 2

1. Créez une classe 'CObjet'
2. Créez les membres privés de type double : masse, vitesse, ec (énergie cinétique)
3. Créez les accesseurs, seul ec doit être en lecture seule.
4. La masse ne peut pas être nulle
5. La vitesse ne peut pas être nulle
6. Créez une méthode void calculer(void)
7. Cette méthode doit calculer l'énergie cinétique d'un objet
8. Le main doit comporter une méthode void comparer(CObjet,CObjet) qui doit comparer l'énergie cinétique de deux objets et qui déterminera et affichera quel est l'objet qui a la plus grande valeur cinétique
9. Utilisez le main pour créer des objets et comparer leurs valeurs.

### Exercice 3

1. Reprenez l'exercice 2
2. Ajouter à la classe 'Clobjet' : un constructeur par défaut, un constructeur paramétré, un constructeur de copie et une méthode privée d'initialisation de l'objet.
3. Dans le main créez un objet o1 sans argument. Créez un objet o2(2,2). Créez un objet dynamique o3. Créez un objet dynamique o4. Allouez de la mémoire pour o3(4,32). Allouez de la mémoire pour o4 en construisant cet objet avec l'objet o3. Calculez l'énergie de o1. Calculez l'énergie de o2. Comparez o1 et o2. Calculez l'énergie de o3. Calculez l'énergie de o4. Comparez o3 et o4. Libérez la mémoire pour o3. Affichez la masse et la vitesse de o4.

### Exercice 4



1. Veuillez prendre connaissance du diagramme de classe et de séquence.
2. Vous devez coder le programme qui répond à ces deux diagrammes
3. Indications supplémentaires :
  - a. Chaque voiture comporte un ID. Cet ID est généré automatiquement par une méthode statique.
  - b. L'accélération et la vitesse ne peuvent être nuls.
  - c. L'accélération est départ arrêté. Le véhicule va accélérer jusqu'à atteindre sa vitesse maximale. L'affichage de la vitesse aura lieu tout au long de l'accélération. Lorsque

le véhicule aura atteint sa vitesse maximale un coefficient de performance sera affiché (Vitesse maximale/Accélération).

4. Pour le main :
  - a. Créez une voiture v1(20,1)
  - b. Créez une voiture v2(300,10)
  - c. Déclarez une voiture dynamique v3
  - d. Allouez de la mémoire pour v3(160,2)
  - e. Faites accélérer les voitures.

# CORRECTION

```
Q1-----
#pragma once
class CLa
{
public :
    void afficher(void);
};

#include "CLa.h"
void CLa::afficher(void)
{
    cout << "Bonjour" << endl;
}

#include "pch.h"
#include "CLa.h"
#include <iostream>;

using namespace System;
int main(array<System::String ^> ^args)
{
    CLa o1;
    CLa* o2;

    o2 = new CLa();

    o1.afficher();
    o2->afficher();

    std::cout << "Adresse de o1 : " << &o1 << std::endl;
    std::cout << "Adresse de o2 : " << &o2 << std::endl;
    std::cout << "Adresse pointee par o2 : " << o2 << std::endl;
    std::cout << "Taille type : " << sizeof(CLa) << " Octets" << std::endl;
    std::cout << "Taille o1 : " << sizeof(o1) << " Octets" << std::endl;
    std::cout << "Taille o2 : " << sizeof(o2) << " Octets" << std::endl;

    delete o2;

    return 0;
}
```

Q2-----

```
#pragma once
class CObjet
{
private :
    double masse;
    double vitesse;
    double ec;

public :
    void calculer(void);
    double getMasse(void);
    double getVitesse(void);
    double getEc(void);
    void setMasse(double);
    void setVitesse(double);
};

#include "pch.h"
#include "Cl.h"
#include <iostream>
#include <math.h>

using namespace std;

void CObjet::calculer(void)
{
    this->ec = 0.00;
    this->ec = 0.5*this->getMasse()*pow(this->getVitesse(),2);

    cout << "Resultat : " << this->ec << endl;
}
double CObjet::getMasse(void)
{
    return this->masse;
}
double CObjet::getVitesse(void)
{
    return this->vitesse;
}
void CObjet::setMasse(double masse)
{
    if (masse < 0)
    {
        this->masse = 0;
    }
    else
    {
        this->masse = masse;
    }
}
void CObjet::setVitesse(double vitesse)
{
    if (vitesse < 0)
    {
        this->vitesse = 0;
    }
    else
    {
        this->vitesse = vitesse;
    }
}
double CObjet::getEc(void)
{
    return this->ec;
}

#include "pch.h"
#include "Cl.h"
#include <iostream>
```

# PROGRAMMATION ORIENTEE OBJET

## 1 – CORBEILLE

```
using namespace System;

void comparer(CLobjet, CLobjet);
int main(array<System::String ^> ^args)
{
    CLobjet obj1;
    CLobjet obj2;

    obj1.setMasse(100);
    obj1.setVitesse(10);
    obj1.calculer();

    obj2.setMasse(10);
    obj2.setVitesse(100);
    obj2.calculer();

    comparer(obj1, obj2);

    return 0;
}

void comparer(CLobjet o1, CLobjet o2)
{
    if (o1.getEc() > o2.getEc())
    {
        std::cout << "L objet 1 a plus d'energie que l'objet 2" << std::endl;
    }
    else
    {
        std::cout << "L objet 2 a plus d'energie que l'objet 1" << std::endl;
    }
}
```

Q3-----

```
#pragma once
class CLobjet
{
private :
    double masse;
    double vitesse;
    double ec;
    void ini_obj(double, double);

public :
    CLobjet(void);
    CLobjet(double, double);
    CLobjet(CLobjet&);
    void calculer(void);
    double getMasse(void);
    double getVitesse(void);
    double getEc(void);
    void setMasse(double);
    void setVitesse(double);
};

#include "pch.h"
#include "CLa.h"
#include <iostream>
#include <math.h>

using namespace std;

CLobjet::CLobjet(void)
{
    this->ini_obj(0.00, 0.00);
}

CLobjet::CLobjet(double vitesse, double masse)
{
}
```

```
        this->ini_obj(vitesse, masse);
    }
    Clobjet::Clobjet(Clobjet& obj)
    {
        this->ini_obj(obj.getVitesse(), obj.getMasse());
    }
    void Clobjet::ini_obj(double vitesse, double masse)
    {
        this->setVitesse(vitesse);
        this->setMasse(masse);
    }
    void Clobjet::calculer(void)
    {
        this->ec = 0.00;
        this->ec = 0.5*this->getMasse()*pow(this->getVitesse(),2);

        cout << "Resultat : " << this->ec << endl;
    }
    double Clobjet::getMasse(void)
    {
        return this->masse;
    }
    double Clobjet::getVitesse(void)
    {
        return this->vitesse;
    }
    void Clobjet::setMasse(double masse)
    {
        if (masse < 0)
        {
            this->masse = 0;
        }
        else
        {
            this->masse = masse;
        }
    }
    void Clobjet::setVitesse(double vitesse)
    {
        if (vitesse < 0)
        {
            this->vitesse = 0;
        }
        else
        {
            this->vitesse = vitesse;
        }
    }
    double Clobjet::getEc(void)
    {
        return this->ec;
    }
    #include "pch.h"
    #include "Cla.h"
    #include <iostream>

    using namespace System;

    void comparer(Clobjet,Clobjet);
    void comparer(Clobjet*, Clobjet*);

    int main(array<System::String ^> ^args)
    {
        Clobjet o1;
        Clobjet o2(2,2);
        Clobjet* o3;
        Clobjet* o4;

        o3 = new Clobjet(4, 32);
```

# PROGRAMMATION ORIENTEE OBJET

## 1 – CORBEILLE

```
    o4 = new CObjet(*o3);
    o1.calculer();
    o2.calculer();
    comparer(o1, o2);

    o3->calculer();
    o4->calculer();
    comparer(o3, o4);

    delete o3;
    std::cout << o4->getMasse() << " - " << o4->getVitesse() << std::endl;

    return 0;
}

void comparer(CObjet obj1, CObjet obj2)
{
    if (obj1.getEc() > obj2.getEc())
    {
        std::cout << "L objet 1 a plus d'energie que l'objet 2" << std::endl;
    }
    else
    {
        std::cout << "L objet 2 a plus d'energie que l'objet 1" << std::endl;
    }
}

void comparer(CObjet* obj1, CObjet* obj2)
{
    if (obj1->getEc() > obj2->getEc())
    {
        std::cout << "L objet 1 a plus d'energie que l'objet 2" << std::endl;
    }
    else
    {
        std::cout << "L objet 2 a plus d'energie que l'objet 1" << std::endl;
    }
}
```

Q4-----

```
class CLvoiture
{
private :

    int vitesseMax;
    int vitesse;
    int acceleration;
    int idV;

    void ini_obj(int,int);

public :
    static int CLvoiture::idVoiture;
    CLvoiture(void);
    CLvoiture(int,int);
    CLvoiture(CLvoiture&);
    void accelerer(void);
    int getAcceleration();
    int getVitesseMax();
    int getID();
    void setAcceleration(int);
    void setVitesseMax(int);
};

#include "pch.h"
#include "CLa.h"
#include <iostream>
#include <math.h>
```



```
using namespace std;

int CLvoiture::idVoiture = 0;
CLvoiture::CLvoiture(void)
{
    this->ini_obj(0,0);
}
CLvoiture::CLvoiture(int vitesseMax, int acceleration)
{
    this->ini_obj(vitesseMax, acceleration);
}
CLvoiture::CLvoiture(CLvoiture& voiture)
{
    this->ini_obj(voiture.getVitesseMax(), voiture.getAcceleration());
}
void CLvoiture::ini_obj(int vitesseMax, int acceleration)
{
    CLvoiture::idVoiture++;
    this->vitesse = 0;
    this->idV = CLvoiture::idVoiture;
    this->setVitesseMax(vitesseMax);
    this->setAcceleration(acceleration);
    cout << "Creation de la voiture : " << this->getID() << endl;
}
void CLvoiture::accelerer(void)
{
    int i;
    double performance;

    for (i = 0; this->vitesse < this->getVitesseMax(); i++)
    {
        cout << "La voiture : " << this->getID() << " roule a la vitesse de : " << this->vitesse << " km/h en " << i << " unite
de temps." << endl;
        this->vitesse = this->vitesse + this->getAcceleration();
    }

    performance = ((double)this->getVitesseMax()) / ((double)this->getAcceleration());
    cout << "La performance de la voiture " << this->getID() << " est de : " << performance << endl;
}
int CLvoiture::getAcceleration()
{
    return this->acceleration;
}
int CLvoiture::getVitesseMax()
{
    return this->vitesseMax;
}
int CLvoiture::getID()
{
    return this->idV;
}
void CLvoiture::setAcceleration(int acceleration)
{
    if (acceleration > 0)
    {
        this->acceleration = acceleration;
    }
    else
    {
        this->acceleration = 1;
    }
}
void CLvoiture::setVitesseMax(int vitesseMax)
{
    if (vitesseMax > 0)
    {
        this->vitesseMax = vitesseMax;
    }
    else
    {

```

# PROGRAMMATION ORIENTEE OBJET

## 1 – CORBEILLE

```
        this->vitesseMax = 1;
    }
}

#include "pch.h"
#include "CLa.h"
#include <iostream>

using namespace System;

int main(array<System::String ^> ^args)
{
    CLvoiture v1(20,1);
    CLvoiture v2(300, 10);
    CLvoiture* v3;

    v3 = new CLvoiture(160, 2);

    v1.accelerer();
    v2.accelerer();
    v3->accelerer();

    return 0;
}
```