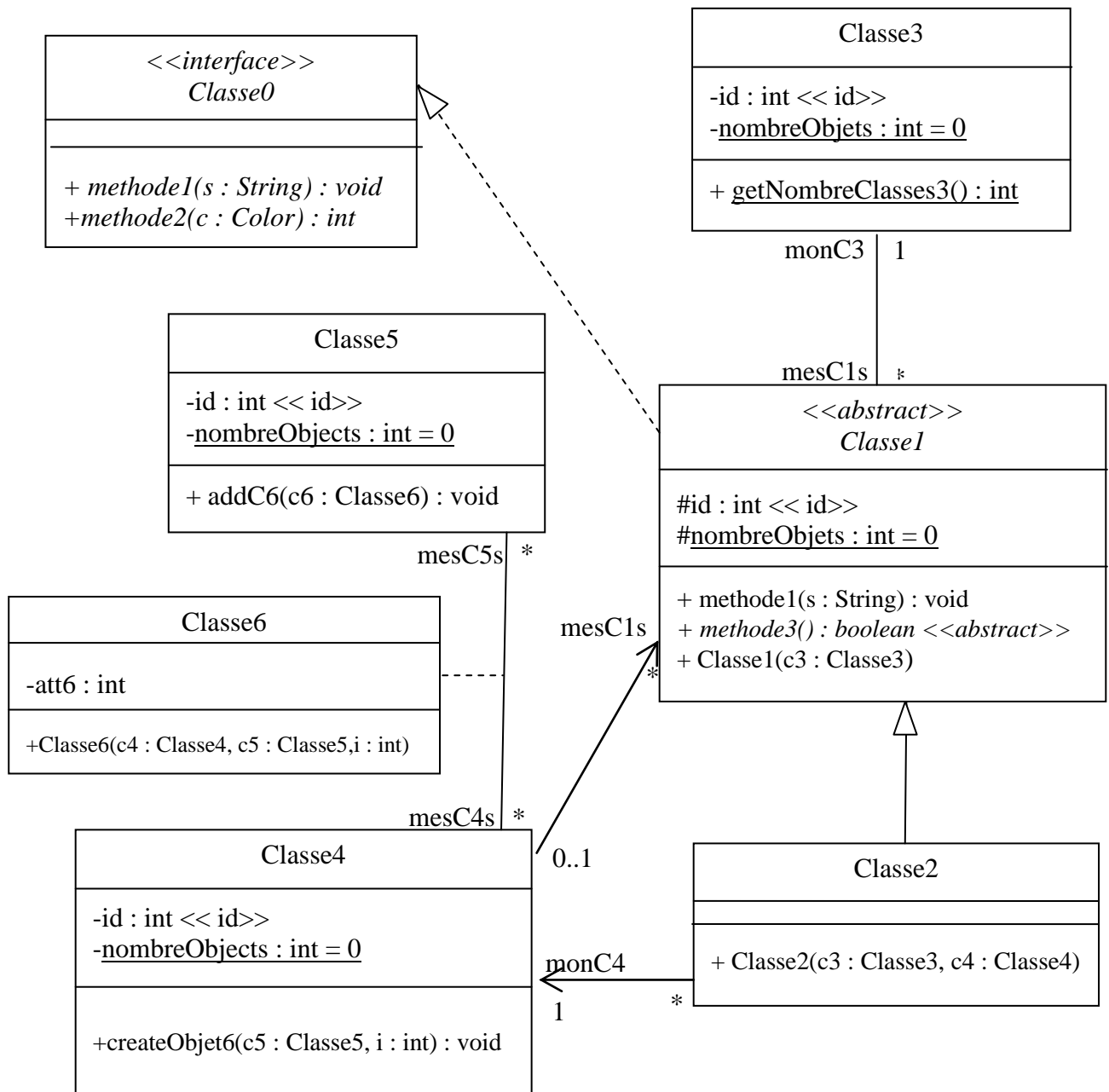


UML - pense-bête : du Diagramme de Classes au Programme C++



```

#ifndef CLASSE3_H
#define CLASSE3_H

using namespace std;

class Classe3 {
public:
    Classe3(){ id = Classe3::nombreObjets; Classe3::nombreObjets++; }
    virtual ~Classe3(){}
    static int getNombreClasses3() {return nombreObjets;};
private:
    int id;
    static int nombreObjets;};

#endif // CLASSE3_H

```

```

#ifndef CLASSE5_H
#define CLASSE5_H

#include <vector>
class Classe6;

class Classe5 {
public:
    Classe5(){ id = Classe5::nombreObjets; Classe5::nombreObjets++;
              mesC6s = vector<Classe6*>();}
    virtual ~Classe5(){}
    void addC6(Classe6* c6) {mesC6s.push_back(c6);}
private:
    int id;
    static int nombreObjets;
    vector<Classe6*> mesC6s;
};

#endif // CLASSE5_H

```

```

#ifndef CLASSE6_H
#define CLASSE6_H

#include "Classe5.h"
class Classe4;

class Classe6 {
public:
    Classe6(){att6 = 0 ; monC4 = NULL; monC5 = NULL;}
    Classe6(Classe4* c4, Classe5* c5, int i) { monC4 = c4; monC5 = c5; att6 = i;}
    virtual ~Classe6(){}
private:
    int att6;
    Classe4* monC4;
    Classe5* monC5;
};

#endif // CLASSE6_H

```

```

#ifndef CLASSE0_H
#define CLASSE0_H

```

```

#include <SFML/Graphics/Color.hpp>

using namespace std;

class Classe0 {
public:
    Classe0(){};
    virtual ~Classe0(){}
    virtual void methode1(string s)=0;
    virtual void methode2 (sf::Color *c)=0;
};

#endif // CLASSE0_H



---



#ifndef CLASSE1_H
#define CLASSE1_H

#include "classe0.h"
#include "classe3.h"

class Classe1 : public Classe0 {
private:
    int id;
    static int nombreObjets;
    Classe3 *monC3;
public:
    Classe1(){
        id = Classe1::nombreObjets;
        Classe1::nombreObjets++; monC3 = NULL; }
    Classe1(Classe3* c3){
        monC3 = c3; id = nombreObjets; nombreObjets++; }
    virtual ~Classe1(){};
    void methode1(string s) {};
    virtual bool methode3() = 0;
};

#endif // CLASSE1_H



---



```

```

#ifndef CLASSE4_H
#define CLASSE4_H

#include <vector>
#include "Classe1.h"
#include "Classe6.h"

class Classe4 {
public:
    Classe4(){
        id = Classe4::nombreObjets; Classe4::nombreObjets++;
        mesC6s = vector<Classe6*>(); mesC1s = vector<Classe1*>(); }
    virtual ~Classe4() {}
    void createObjet6(Classe5* c5, int i) {
        Classe6* c6; c6 = new Classe6(this, c5,i);
        mesC6s.push_back(c6); }
private:
    int id;
    static int nombreObjets;
    vector<Classe6*> mesC6s;
    vector<Classe1*> mesC1s;
};

#endif // CLASSE4_H

```

```

#ifndef CLASSE2_H
#define CLASSE2_H

#include "Classe1.h"
#include "Classe4.h"

class Classe2 : public Classe1 {
public:
    Classe2(){ }
    Classe2(Classe3* c3, Classe4* c4): Classe1(c3) { monC4 = c4; }
    virtual ~Classe2(){}
    bool methode3() { }
    void methode2 (sf::Color *c) { }
private:
    Classe4 *monC4;
};

#endif // CLASSE2_H

```

```

#include <stdlib.h>
#include "Classe2.h"

int Classe1::nombreObjets =0;
int Classe3::nombreObjets =0;
int Classe4::nombreObjets =0;
int Classe5::nombreObjets =0;

int main( int argc, const char* argv[] ){
    Classe2 c2;
    return EXIT_SUCCESS;}

```