

## Héritage en C++: les animaux

classe mère: Pangolin.hpp

```
1 #include <string>
2
3 class Pangolin {
4     int nb_ecaillles_;
5
6 public:
7     const std::string nom_;
8
9     Pangolin(std::string nom, int nb_ecaillles) : nom_(nom), nb_ecaillles_(nb_ecaillles) {}
10    ~Pangolin() = default;
11
12    int get_nb_ecaillles() const { return nb_ecaillles_; }
13    void set_nb_ecaillles(int nb) { this->nb_ecaillles_ = nb; }
14
15    const std::string cri() const { return "Gwark Rhââgn Bwikk"; /* Cri du pangolin */ }
16};
```

classe fille: PangolinALongueQueue.hpp

```
1 #include "Pangolin.hpp"
2
3 class PangolinALongueQueue : public Pangolin {
4     double lgr_queue_;
5
6 public:
7     PangolinALongueQueue(std::string nom, int nb_ecaillles, double lgr)
8         : Pangolin(nom, nb_ecaillles), lgr_queue_(lgr) {}
9     double get_longueur() const { return lgr_queue_; }
10};
```

usage

```
1 #include "PangolinALongueQueue.hpp"
2 #include <iostream>
3
4 int main() {
5     Pangolin p("toto", 2241);
6     PangolinALongueQueue q("tutu", 3321, 42);
7     std::cout << p.nom_ << "{ecaillles = " << p.get_nb_ecaillles() << "; cri= " << p.cri() << "}\n";
8     std::cout << q.nom_ << "{ecaillles = " << q.get_nb_ecaillles() << "; cri= " << q.cri() << ";
9         << "lgr:" << q.get_longueur() << "}\n";
10    return 0;
11};
```

affichage lors de l'usage

```
1 $ clang++ MainPangolin.cpp -o pangolin && ./pangolin
2 toto {ecaillles = 2241; cri= Gwark Rhââgn Bwikk}
3 tutu {ecaillles = 3321; cri= Gwark Rhââgn Bwikk; lgr:42}
```

## Héritage multiple en diamant

```
1 class Base {};
```

```
2 class Deriv1 : public virtual Base {};
```

```
3 class Deriv2 : public virtual Base {};
```

```
4 class Fille : public Deriv1, public Deriv2 {};
```

## Opérateurs de conversion

```
1 struct A {
2     A(int) { } // converting ctor
3     A(int, int) { } // converting ctor
4     operator bool() const { return true; }
5};
```

```
1 A a1 = 1; // copy-init
2 A a2(2); // direct-init
3 A a3 {4, 5}; // direct-list-init
4 A a4 = {4, 5}; // copy-list-init
5 A a5 = (A)1; // explicit cast
6 if (a1); //
7 bool na1 = a1; // copy-init
8 bool na2 = static_cast<bool>(a1);
```

## Redéfinition de fonctions

Marsouin.hpp

```
1 #include <string>
2
3 class Marsouin {
4     public:
5     virtual const std::string cri();
6 };
7 class MarsouinFou : public Marsouin {
8     public:
9     const std::string cri() override;
10};
```

Marsouin.cpp

```
1 #include "Marsouin.hpp"
2
3 const std::string Marsouin::cri() {
4     return "Gahbahiii";
5 }
6
7 const std::string MarsouinFou::cri() {
8     return std::string("Kweghou ") + Marsouin::cri();
9 }
```