

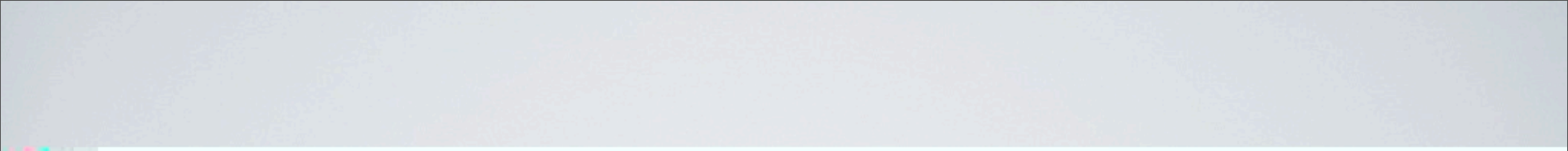
```
class Magnetophone {
    public:
        void demarreEcoute() { ... };
        void demarreEnregistrement() { ... };
};

class Camera {
    public:
        void demarreProjection() { ... };
        void demarreEnregistrement() { ... };
};

class Camescope : private Magnetophone, private Camera {
    public:
        void enregistreFilm() {
            Magnetophone::demarreEnregistrement();
            Camera::demarreEnregistrement();
        }
        void regardeFilm() {
            Magnetophone::demarreEcoute();
            Camera::demarreProjection();
        }
};
```

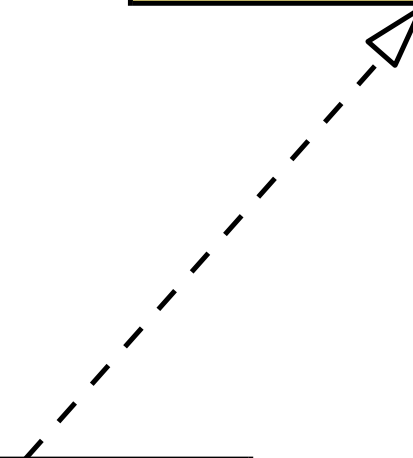
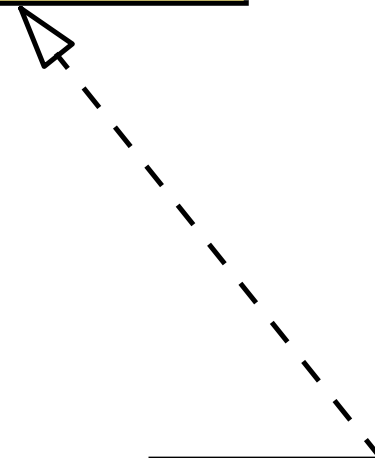
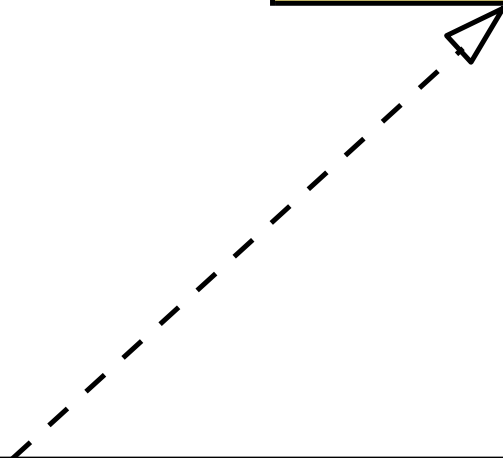
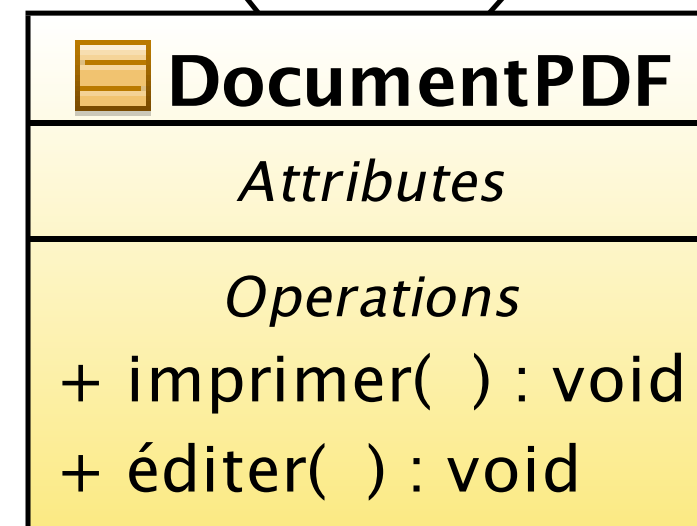
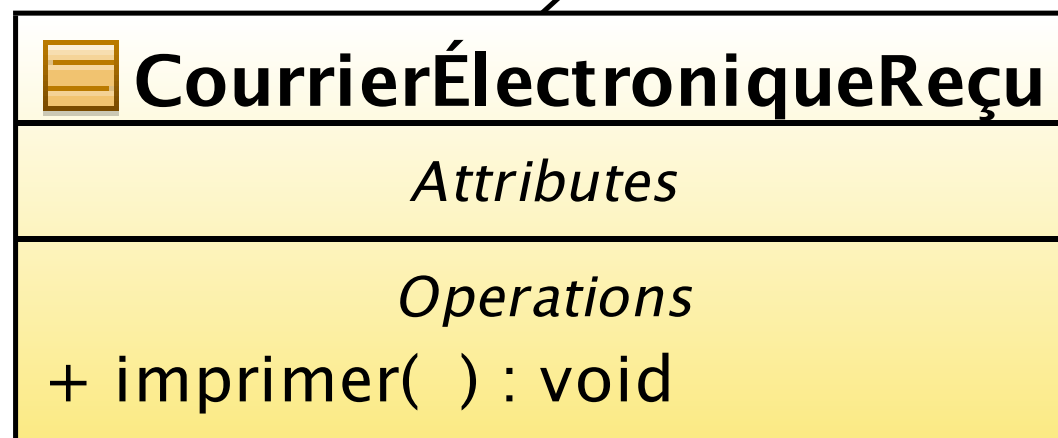
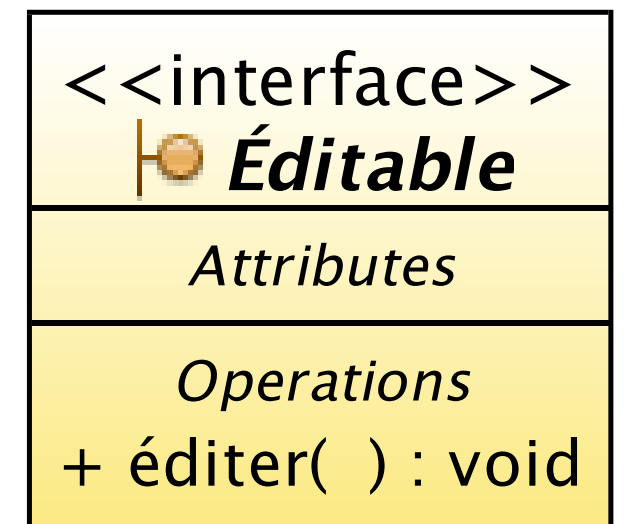
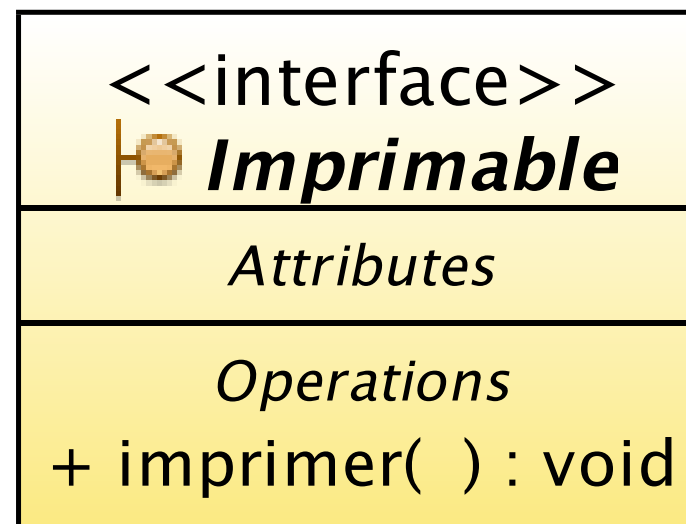


```
class SupportDEnregistrement {};  
  
class Magnetophone : private SupportDEnregistrement {  
    public:  
        void demarreEcoute() { ... };  
        void demarreEnregistrement() { ... };  
};  
  
class Camera : private SupportDEnregistrement {  
    public:  
        void demarreProjection() { ... };  
        void demarreEnregistrement() { ... };  
};  
  
class Camescope : private Magnetophone, private Camera {  
    public:  
        void enregistreFilm() {  
            Magnetophone::demarreEnregistrement();  
            Camera::demarreEnregistrement();  
        }  
        void regardeFilm() {  
            Magnetophone::demarreEcoute();  
            Camera::demarreProjection();  
        }  
};
```



*cascade*



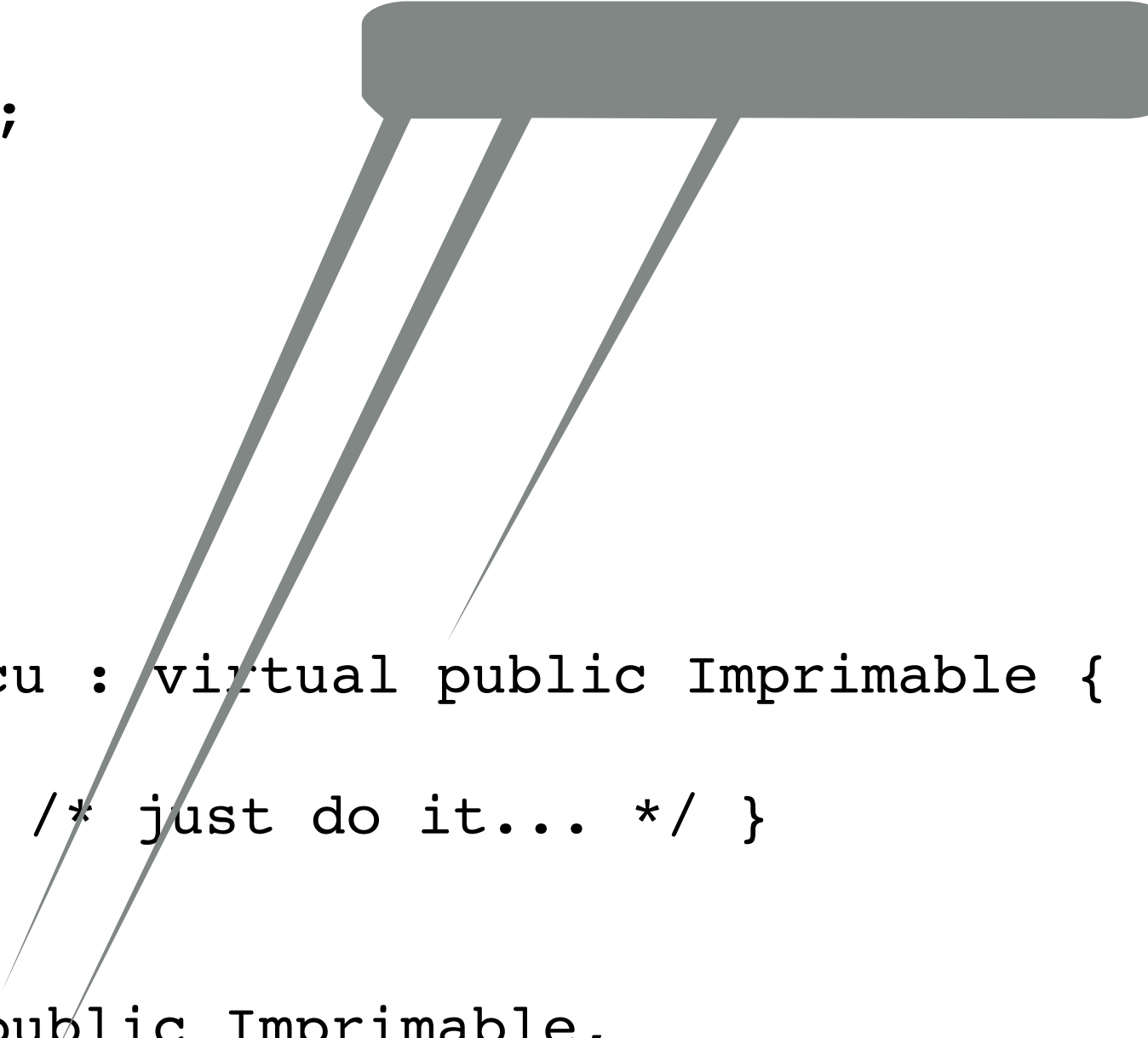


```
class Imprimable {
    public:
        virtual void imprimer()=0;
};

class Editable {
    public:
        virtual void editer()=0;
};

class CourrierElectroniqueRecu : virtual public Imprimable {
    public:
        virtual void imprimer() { /* just do it... */ }
};

class DocumentPDF : virtual public Imprimable,
                    virtual public Editable {
    public:
        virtual void editer() { /* do your best... */ }
        virtual void imprimer() { /* merci Johannes Gensfleisch
                                   zur Laden zum Gutenberg */ }
};
```

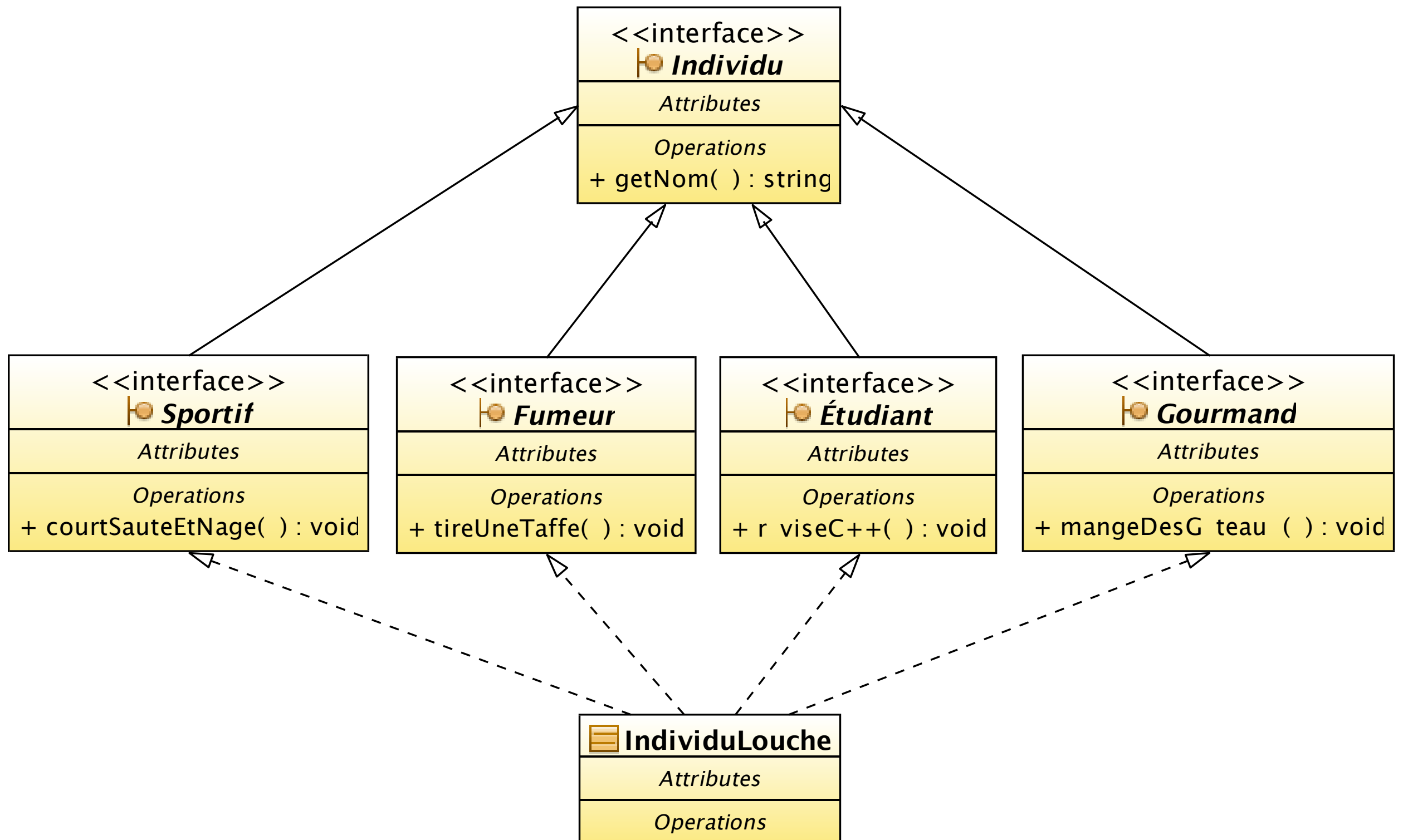


*héríte*

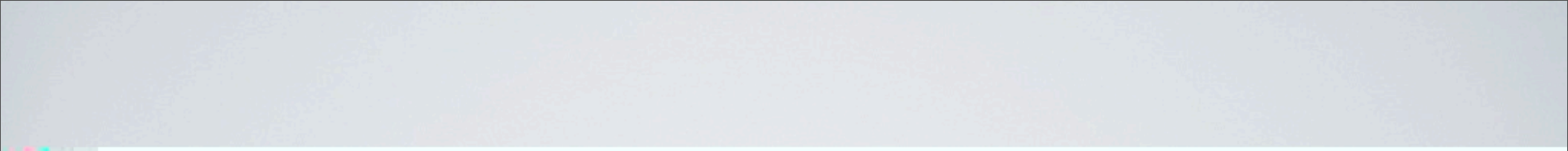
*i.e.*

```
class IPoint {  
    public:  
        virtual void affiche()=0;  
};  
  
class ICouleur {  
    public:  
        virtual void affiche()=0;  
};  
  
class PointColore : virtual public IPoint,  
                    virtual public ICouleur{  
    public:  
        virtual void affiche() { /* OK */ }  
};
```

affiche



```
class Base {  
};  
  
class SousClasse1 : virtual public Base {  
};  
  
class SousClasse2 : virtual public Base {  
};  
  
class ToutEnBas :  
    virtual public SousClasse1,  
    virtual public SousClasse2 {  
};
```



```
class Base {
    public:
        Base(int v) {}
};

class Classe1 : virtual public Base {
    public:
        Classe1() : Base(3) {}
};

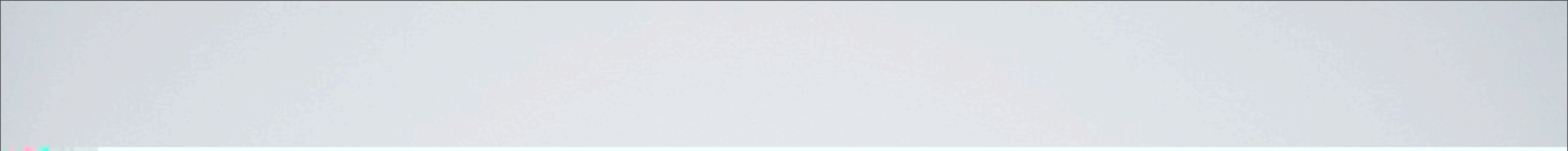
class Classe2 : virtual public Base {
    public:
        Classe2() : Base(5) {}
};

class ToutEnBas : virtual public Base,
                  virtual public Classe1,
                  virtual public Classe2 {
    public:
        ToutEnBas() : Base(4), Classe1(), Classe2() {}
};
```

Base

Base





```
class Classe1 {
    public:
        virtual void f() {};
};

class Classe2 {
    public:
        virtual void g() {};
};

class T : public Classe1, public Classe 2 {
};

int main() {
    T unT;
    Classe1 *p1;
    p1 = &unT; // OK!
    Classe2 *p2;
    p2 = &unT; // OK!
    p1 = p2; // Non...
    p1 = dynamic_cast<Classe1 *>(p2); // oui si Classe1 polymorphe
    return 0;
}
```

