

Support de cours

Cours:

## Introduction à la programmation orientée objet (en C++)

Vidéo:

### W15-01-intropolym-CPP-pt2

Concepts (extraits des sous-titres générés automatiquement) :

**Choix de la méthode. Type de résolution. Résolution dynamique des liens. Fonction de l'objet. Petit exemple du début. Méthodes virtuelles. Premier est. Résolution statique des liens. Instances réelles des objets. Étiquette particulière. Variable de type personnage. Résolution dynamique. Travers de référence. Ingrédients particuliers. Résolution statique.**



[vers la recherche de séquences vidéo](#)

(dans Introduction à la programmation orientée objet (en C++).)



[vers la vidéo](#)

Center for Digital Education. Plus de matériel de soutien pédagogique ici :

<https://www.epfl.ch/education/educational-initiatives/cede/educational-technologies-gallery/boocs-en/>

# Polymorphisme et résolution dynamique des liens (Partie 2)

## Introduction à la programmation orientée objet (en C++)

Jean-Cédric Chappelier, Jamila Sam et Vincent Lepetit

...

notes

résumé

0m 0s



## Résolution des liens (2)

En C++, c'est le **type de la variable** qui **détermine la méthode** à exécuter :

☞ résolution **statique** des liens

```
class Personnage {
public:
    // ...
    void rencontrer(Personnage& p) const
    { cout << "Bonjour !" << endl; }
};
```

```
class Guerrier : public Personnage {
public:
    // ...
    void rencontrer(Personnage& p) const
    { cout << "Boum !" << endl; }
};
```

```
void faire_rencontrer(
    Personnage const& un,
    Personnage const& autre) {
    cout << un.get_nom() << " rencontre "
        << autre.get_nom() << " : ";
    un.rencontrer(autre);
}
```

```
int main() {
    Guerrier g;
    Voleur v;

    faire_rencontrer(g, v);
}
```

Dans certains cas, il peut cependant sembler plus naturel d'avoir recours à un autre type de résolution, ce que l'on appelle la résolution dynamique des liens. Une résolution où le choix de la méthode va se faire à l'exécution ou dynamiquement en fonction de l'objet réellement contenu dans la variable, ce qui voudrait dire ici que l'on pourrait faire en sorte que la méthode rencontrer soit celle de l'objet réellement stocké dans la variable de type Personnage. La résolution statique des liens est ce qui se met en œuvre

notes

résumé

0m 1s



Il pourrait dans certains cas sembler plus naturel de choisir la méthode correspondant à la *nature réelle de l'instance*.

Dans ces cas, il faut permettre la **résolution dynamique des liens** :

- Le *choix de la méthode* à exécuter se fait à *l'exécution*, en fonction de la *nature réelle des instances*

2 ingrédients pour cela :

**références/pointeurs**

et

**méthodes virtuelles**

par défaut en C++ si l'on ne fait rien de particulier. Il est cependant possible d'avoir recours à la résolution dynamique des liens mais pour cela, il faut faire intervenir deux ingrédients particuliers. Le premier est qu'il faut que les méthodes que l'on souhaite voir résolues dynamiquement soient déclarées avec une étiquette particulière, que ce soit des méthodes virtuelles et il faut qu'elle puisse s'exercer sur les instances réelles des objets via des références ou des pointeurs. Notre petit exemple du début fait donc de la résolution statique des liens ici. Pour qu'il fasse de la résolution dynamique, il faudrait que la méthode rencontrée soit virtuelle et il faudrait qu'elle s'applique non pas à un objet mais à une référence à cet objet ou à un pointeur vers cet objet. Ce qui n'est pas le cas ici. Nous avons bel et bien ici de la résolution statique. Les méthodes virtuelles et leur utilisation au travers de référence et de pointeur vont faire l'objet de notre attention dans la séquence suivante. dans la séquence suivante.

notes

résumé

0m 37s

