--,--...-

Concente

Etude de cas

Programmation Orientée Objet : Polymorphisme, 1^{re} partie

Jean-Cédric Chappelier

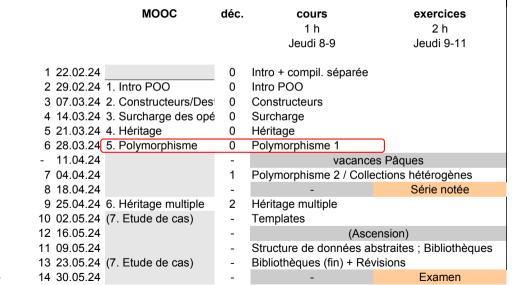
Laboratoire d'Intelligence Artificielle Faculté I&C





Concepts
Etude de cas

Organisation du travail (semestre)



Objectifs Concepts

Objectifs de la leçon d'aujourd'hui

Etude de cas

- Concepts fondamentaux
- Étude de cas

Concepts fondamentaux

- notion de polymorphisme
 - 1. notion FONDAMENTALE!

```
Attention aux « tests de type » : JAMAIS!!!
```

- 2. en C++: 2 ingrédients: ...
- méthodes virtuelles : virtual (transmis par héritage; mais je vous conseille de le remettre à chaque fois pour mémoire) override (optionnel, mais conseillé)
- méthodes virtuelles pures et classes abstraites
 - 1. virtual plus = 0
 - 2. on ne peut pas créer d'instance de classe abstraite!
- (semaine prochaine : collections héterogènes)

Polymorphismes

Concepts

En programmation, on distingue deux types de polymorphismes :

- le polymorphisme des traitements (ou ad hoc)
 - mécanisme de surcharge des fonctions/méthodes : le même identificateur est utilisé pour désigner des séguences d'instructions différentes
- le polymorphisme des données (ou universel)
 - le polymorphisme d'inclusion :
 - le même code peut être appliqué à des données de types différents liés entre eux par une relation de sous-typage
 - hiérarchies de classes
 - le polymorphisme paramétrique :
 - le même code peut être appliqué à n'importe quel type (généricité)
 - Cours sur les *templates* dans quelques semaines

```
Objectils
```

Concepts

Etude de cas

```
Exemple illustratif: l'affichage polymorphique
```

```
class Affichable {
public:
 virtual void affiche(ostream& flot) const = 0;
};
ostream& operator<<(ostream& flot, const Affichable& objet)
  objet.affiche(flot);
 return flot;
class Machin: public Affichable
public:
// ...
  virtual void affiche(ostream& flot) const override {
   // ...
```

Etude de cas

Série notée 2019 (sujet disponible sur le Moodle du cours)