

Programmation Orientée Objet

Héritage

Pr Imane DAOUDI

2016-2017

Plan

1. Introduction à la programmation orientée objet
2. Généralités
3. Opérateurs sur tableaux, pointeurs et objets
4. Les fonctions
5. Programmation des classes et objets
6. Appels et surcharges
7. L' héritage

7. Héritage

➤ Définition.

- Relation de **spécialisation/généralisation** entre deux classes. Elle indique qu'une classe dite **classe fille spécialise** une autre classe dite **classe mère**, i.e. qu'elle possède les attributs et les méthodes de la classe mère **plus** d'autres qui lui sont propres
- Rq: Un attribut ou une méthode sont dits **protégés** si leur utilisation est limitée à la classe et ses descendantes.

7. Héritage

➤ Déclaration d'une classe fille

Syntaxe : (après avoir déclaré une classe « mère »)

class fille : *visibilité* mere

{

Private :

... // déclaration des nouveaux membres privés

Protected :

... // déclaration des nouveaux membres protégés

Public :

... // déclaration des nouveaux membres publics

};

- Le mot clé **visibilité** permet de définir le statut dans la classe fille des membres hérités de la classe mère.
- Il peut prendre trois valeurs : **public**, **private** ou **protected**.

7. Héritage

➤ Déclaration d'une classe fille

Récapitulatifs des visibilités

Type d'héritage	Visibilité d'un membre dans la classe mère	Visibilité de ce membre dans la classe fille
public	public	public
	protected	protected
	private	private (inaccessible)
protected	public	protected
	protected	protected
	private	private (inaccessible)
private	public	private (accessible)
	protected	private (accessible)
	private	private (inaccessible)

7. Héritage

- Exemple d'héritage public d'un attribut public :
 - Dans la classe « fille », l'attribut « argent » a conservé sa visibilité publique. Il y est donc accessible, et le sera par les descendances futures.

```
class mere
{ public :
    double argent; };
```

```
class fille : public mere
{ public :
    void afficher(void)
    { cout << "argent accessible : " << argent << endl; } };
```

7. Héritage

- Exemple d'héritage public d'un attribut public (**suite**) :
 - Ayant conservé sa visibilité publique, l'attribut « argent » est accessible depuis l'extérieur des classes « mère » et « fille ».

```
int main()
{ mere memere;
  memere.argent = 100000;
  fille fille;
  fille.argent = 34; }
```

7. Héritage

➤ Exemple d'héritage public d'un attribut protected :

- Dans la classe « fille », l'attribut « argent » a conservé sa visibilité protected. Il y est donc accessible, et le sera par les descendances futures.

```
class mere
{ protected :
    double argent; };

class fille : public mere
{ public :
    void afficher(void)
    { cout << "argent accessible : " << argent << endl; } };
```

7. Héritage

- Exemple d'héritage public d'un attribut privé :

```
4 class mere
5 { private :
6   double argent; };
7
8 class fille : public mere
9 { public :
10  void afficher(void)
11  { cout << "argent inaccessible : " << argent << endl; } };
12
```

Dans la classe « fille », l'attribut « argent » est inaccessible.

7. Héritage

- Exemple d'héritage protected d'un attribut public:

```
class mere
{ public :
    double argent; };

class fille : protected mere
{ public :
    void afficher(void)
    { cout << "argent accessible : " << argent << endl; } };
```

Dans la classe « fille », l'attribut « argent » est passé d'une visibilité publique à protected. Il y est donc accessible, et le sera par les descendances futures.

7. Héritage

- Exemple d'héritage protected d'un attribut public (suite):

```
int main()
{ mere memere;
  memere.argent = 100000;
  fille fille;
  fille.argent =34; }
```

File	Line	Message
C:\Users\Laure...		In function 'int main()':
C:\Users\Laure...	6	error: 'double mere::argent' is inaccessible
C:\Users\Laure...	17	error: within this context
=== Build finished: 2 errors, 0 warnings ===		

Par contre, étant devenu protected dans la classe « fille », il n'est plus accessible depuis l'extérieur.

7. Héritage

- Exemple d'héritage private d'un attribut protected

```
class mere
{ protected :
    double argent; };

class fille : private mere
{ public :
    void afficher(void)
    { cout << "argent accessible : " << argent << endl; } };
```

Dans la classe « fille », l'attribut « argent » est devenu private. Etant déclaré protected dans la classe mère il est toutefois accessible depuis la classe « fille ». Mais ne le sera pas par ses descendantes

7. Héritage

- Exemple d'héritage private d'un attribut protected

```
class mere
{ protected :
    double argent; };

class fille : private mere
{ public :
    void afficher(void)
    { cout << "argent accessible : " << argent << endl; } };
```

Dans la classe « fille », l'attribut « argent » est devenu private. Etant déclaré protected dans la classe mère il est toutefois accessible depuis la classe « fille ». Mais ne le sera pas par ses descendantes