Trường Đại học Khoa học Tự nhiên
Khoa Công nghệ Thông tin
Bộ môn Công nghệ Phần mềm

Phân tích và thiết kế phần mềm

Ôn tập về hướng đối tượng

Giảng viên:
TS. Trần Minh Triết – ThS. Đặng Bình Phương

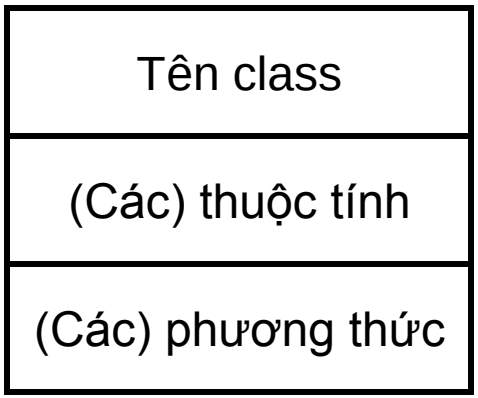
Tham khảo

"Mastering Object-Oriented Analysis and Design with UML 2.0"

IBM Software Group

Nhắc lại về hướng đối tượng

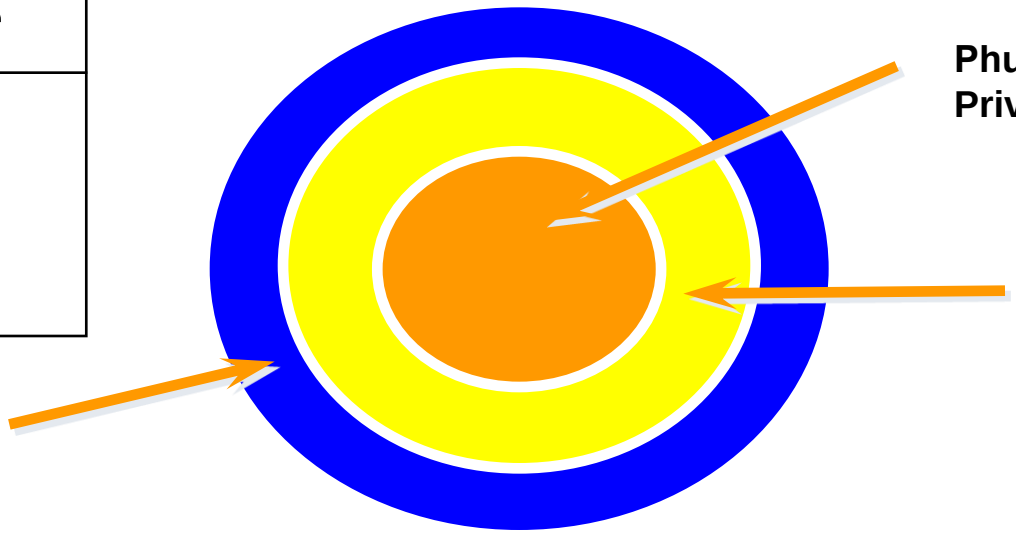
Một số ký hiệu



Public/Protected/Private

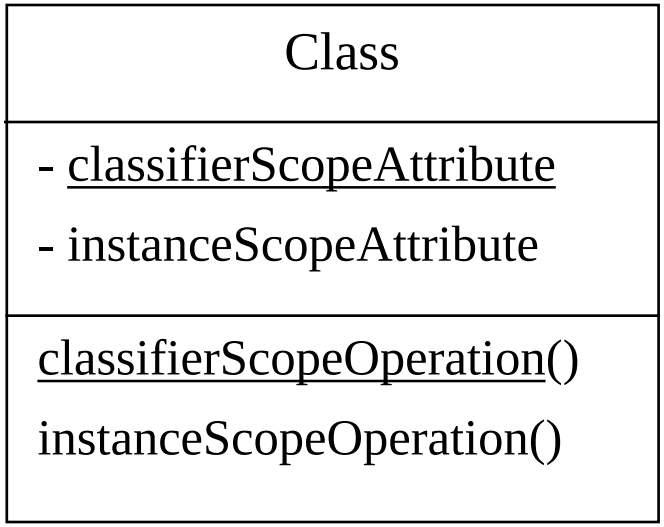
+	Thuộc tính/Phương thức public
#	Thuộc tính/Phương thức protected
-	Thuộc tính/Phương thức private

Class
- privateAttribute # protectedAttribute
+publicOp() # protectedOp() - privateOp()



Tầm vực

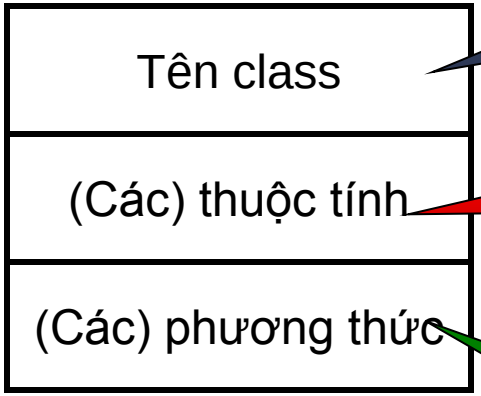
- Xác định số lượng thể hiện của thuộc tính / phương thức



Ví dụ

Student
- name - address - studentID - nextAvailID : int
+ addSchedule(theSchedule : Schedule, forSemester : Semester) + getSchedule(forSemester : Semester) : Schedule + hasPrerequisites(forCourseOffering : CourseOffering) : boolean # passed(theCourseOffering : CourseOffering) : boolean + getNextAvailID() : int

Nhận xét

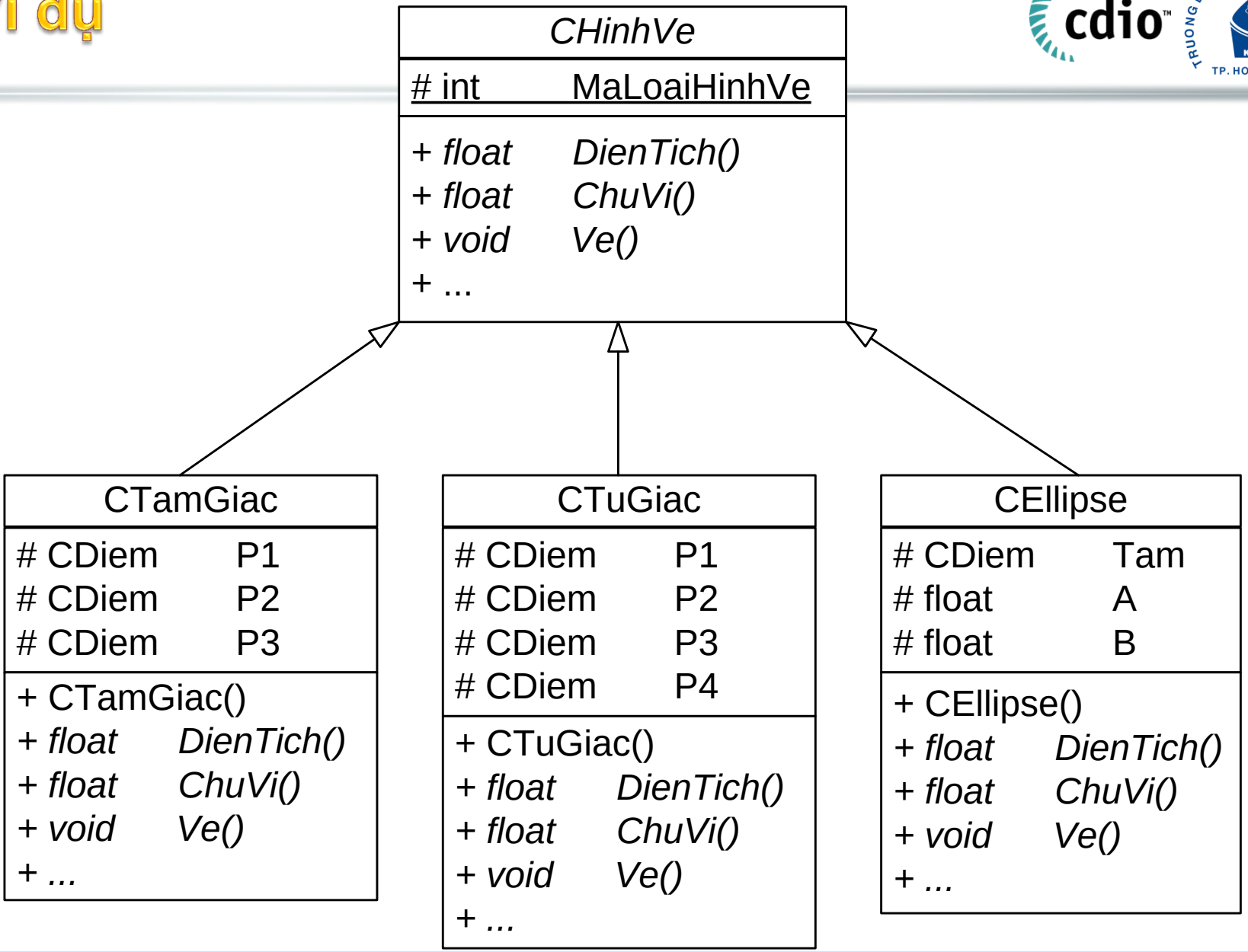


Bình thường: Class bình thường/Interface
In nghiêng: Class thuần ảo
Gạch dưới: Object (không phải class)

Bình thường: Thuộc tính bình thường
In nghiêng: không sử dụng
Gạch dưới: Thuộc tính static

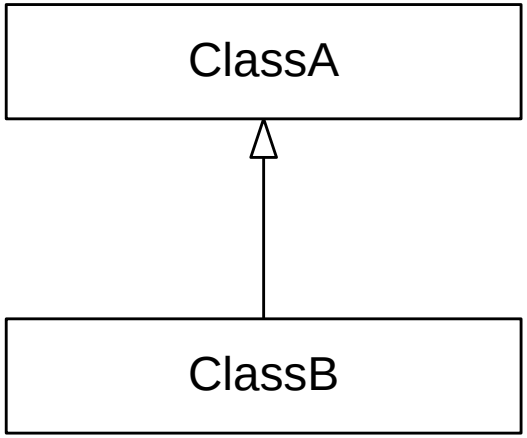
Bình thường: Phương thức bình thường
In nghiêng: Phương thức virtual
Gạch dưới: Phương thức static

Ví dụ


CHinhVe
int MaLoaiHinhVe
+ float DienTich()
+ float ChuVi()
+ void Ve()
+ ...
CTamGiac
CDiem P1
CDiem P2
CDiem P3
+ CTamGiac()
+ float DienTich()
+ float ChuVi()
+ void Ve()
+ ...
CTuGiac
CDiem P1
CDiem P2
CDiem P3
CDiem P4
+ CTuGiac()
+ float DienTich()
+ float ChuVi()
+ void Ve()
+ ...
CEllipse
CDiem Tam
float A
float B
+ CEllipse()
+ float DienTich()
+ float ChuVi()
+ void Ve()
+ ...

Quan hệ giữa các lớp đối tượng

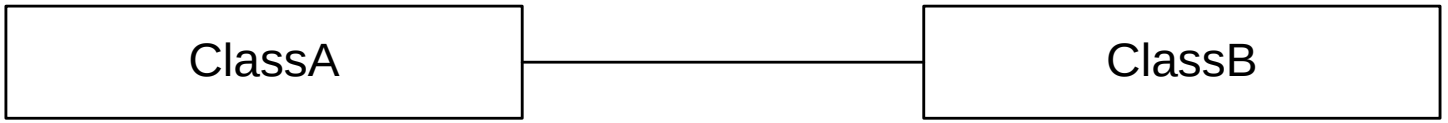
- Quan hệ kế thừa



  - ClassB kế thừa từ ClassA
  - ClassB là một trường hợp đặc biệt của ClassA
  - ClassA là trường hợp tổng quát của ClassB

Quan hệ giữa các lớp đối tượng

- Quan hệ Association

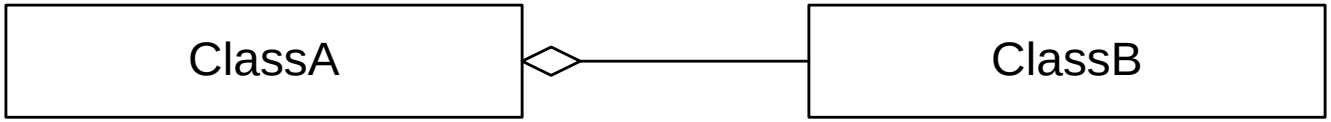


  - Hoặc
    - Trong ClassA có thuộc tính có kiểu là ClassB
  - Hoặc
    - Trong ClassB có thuộc tính có kiểu là ClassA

  - Nhận xét: Về mặt lập trình, thuộc tính có thể được lưu trữ dạng biến đơn, biến mảng, hay biến con trỏ
  - Ví dụ:?

Quan hệ giữa các lớp đối tượng

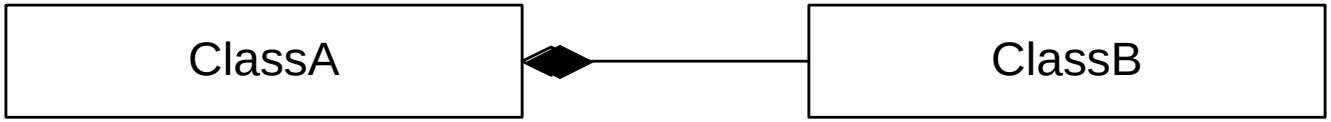
- Quan hệ Aggregation



  - Đã xác định được ClassA và ClassB có quan hệ Association với nhau
  - Xác định rõ hơn:
    - Trong object của ClassA có chứa (trong phần thuộc tính) object của ClassB
    - ObjectX của ClassA bị hủy thì ObjectY của ClassB (bên trong ObjectX) vẫn có thể còn tồn tại
  - Ví dụ:?

Quan hệ giữa các lớp đối tượng

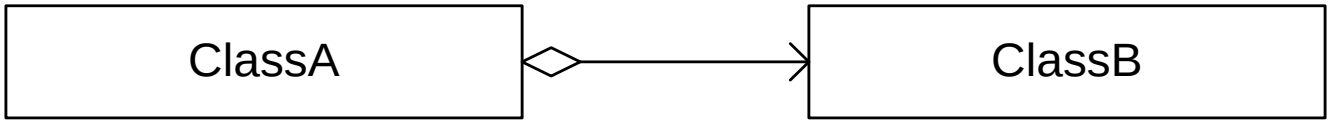
- Quan hệ Composition



  - Đã xác định được ClassA và ClassB có quan hệ Association với nhau
  - Xác định rõ hơn:
    - Trong object của ClassA có chứa (trong phần thuộc tính) object của ClassB
    - ObjectX của ClassA bị hủy thì ObjectY của ClassB (bên trong ObjectX) không thể còn tồn tại
  - Ví dụ:?

Quan hệ giữa các lớp đối tượng

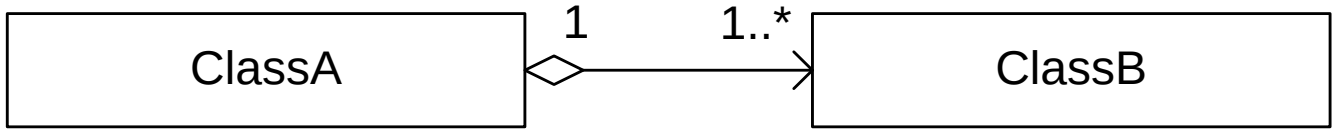
- Chiều của quan hệ (Association, Aggregation, Composition)



  - Nếu quan hệ là 1 chiều: đa số các lời gọi hàm được gọi theo đúng chiều của quan hệ
  - Nếu quan hệ là 2 chiều: không vẽ mũi tên

Quan hệ giữa các lớp đối tượng

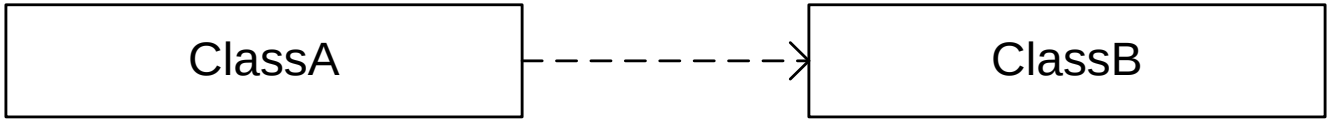
- Bản số - Multiplicity (Association, Aggregation, Composition)



  - Ý nghĩa
  - Ví dụ:
    - 1
    - 2
    - 1..*
    - 0..*
    - *
    - 1, 3, 5..9

Quan hệ giữa các lớp đối tượng

- Quan hệ Dependency



  - ClassA và ClassB không có quan hệ Association
  - ClassA "phụ thuộc" vào ClassB

Tham số truyền vào

```
class A
{
    void    F(B   x)
    {
     …
    }
};
```

Kết quả trả ra

```
class A
{
    B    F()
    {
     …
    }
};
```

Biến cục bộ

```
class A
{
    void    F()
    {
        B    x;
    }
};
```

Trong ClassA có sử dụng biến toàn cục (kiểu B), hoặc sử dụng phương thức/thuộc tính static của ClassB