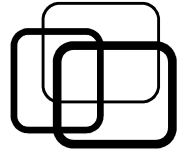


Tổng quan về Mẫu thiết kế HĐT

GV. Nguyễn Minh Huy

cuu duong than cong . com

Nội dung

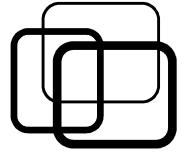


- Ôn tập HĐT.
- Giới thiệu mẫu thiết kế.
- Ký hiệu & thuật ngữ.

cuu duong than cong . com

cuu duong than cong . com

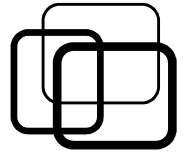
Nội dung



- Ôn tập HĐT.
- Giới thiệu mẫu thiết kế.
- Ký hiệu & thuật ngữ.

cuu duong than cong . com

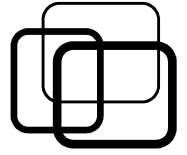
cuu duong than cong . com



- Đối tượng là gì?
- Chương trình hướng đối tượng?
 - Khai báo lớp và sử dụng đối tượng?
→ Chưa đủ!!
 - Cần tuân thủ ba tính chất hướng đối tượng!!

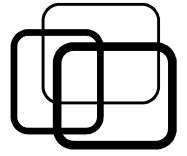


cuu duong than cong . com



- Ba tính chất lập trình hướng đối tượng?
 - Tính đóng gói (encapsulation)
 - ➔ Quy tắc hộp đen.
 - Tính kế thừa (inheritance)
 - ➔ Tái sử dụng cái có sẵn.
 - Tính đa hình.
 - ➔ Hàm ảo + liên kết động.

cuu duong than cong . com



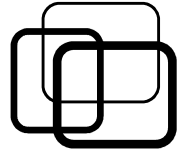
■ Mục đích của lập trình hướng đối tượng?

■ Giảm thời gian và công sức lập trình

- Giảm gắn kết mã nguồn (spaghetti code).
- Tái sử dụng và giảm phụ thuộc cài đặt cụ thể.
- Thích ứng thay đổi, dễ bảo trì, nâng cấp.



Nội dung

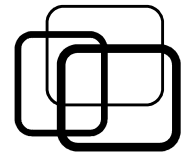


- Ôn tập HĐT.
- **Giới thiệu mẫu thiết kế.**
- Ký hiệu & thuật ngữ.

cuu duong than cong . com

cuu duong than cong . com

Giới thiệu mẫu thiết kế

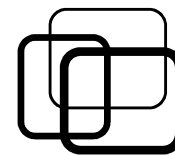


■ Vì sao có mẫu thiết kế?

- Các tính chất hướng đối tượng mơ hồ, khó hiểu.
- Nhu cầu tái sử dụng tri thức.
 - ➔ Cần có những bài giải mẫu.
- Mẫu thiết kế Gang of Four (GoF Patterns):
 - Ra đời tại OOPSLA 1994.
 - 23 bài giải mẫu cho những vấn đề thiết kế thường gặp.

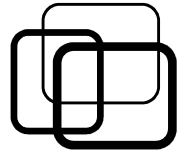


Giới thiệu mẫu thiết kế



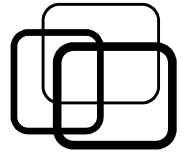
■ Phân loại mẫu thiết kế GoF:

		Purpose		
		Creational	Structural	Behavioral
Scope	Class	<u>Factory Method (107)</u>	<u>Adapter (139)</u>	<u>Interpreter (243)</u> <u>Template Method (325)</u>
	Object	<u>Abstract Factory (87)</u> <u>Builder (97)</u> <u>Prototype (117)</u> <u>Singleton (127)</u>	<u>Adapter (139)</u> <u>Bridge (151)</u> <u>Composite (163)</u> <u>Decorator (175)</u> <u>Facade (185)</u> <u>Proxy (207)</u>	<u>Chain of Responsibility (223)</u> <u>Command (233)</u> <u>Iterator (257)</u> <u>Mediator (273)</u> <u>Memento (283)</u> <u>Flyweight (195)</u> <u>Observer (293)</u> <u>State (305)</u> <u>Strategy (315)</u> <u>Visitor (331)</u>



- Cấu trúc một mẫu thiết kế:
 - Tổng quan (pattern name, summary).
 - Bài toán (motivation).
 - Lời giải:
 - Ngữ cảnh áp dụng (applicability).
 - Sơ đồ lớp (structure).
 - Các thành phần cấu thành (participants).
 - Bàn luận:
 - Ưu, khuyết (consequences).
 - Cách cài đặt hiệu quả (implementation/sample code).
 - Áp dụng thực tế (known uses).

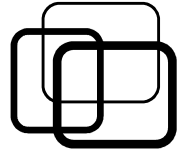
Giới thiệu mẫu thiết kế



- Hướng phát triển:
 - Mẫu thiết kế mới.
 - Mẫu kết hợp (compound patterns).
 - Mẫu nên tránh (anti-patterns, code smells).
 - Mẫu phân tích (analysis patterns).

cuu duong than cong . com

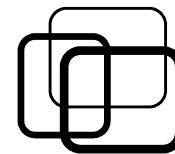
Nội dung



- Ôn tập HĐT.
- Giới thiệu mẫu thiết kế.
- **Ký hiệu & thuật ngữ.**

cuu duong than cong . com

cuu duong than cong . com



■ Ký hiệu UML:

■ Lớp đối tượng:

- Thuộc tính, phương thức.
- Tầm vực.
- Chú thích mã giả.

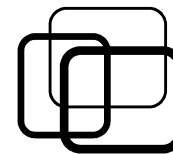
■ Kế thừa (IS-A):

- Reuse vs. Override.
- Hàm thuần ảo.

■ Bao hàm (HAS-A).

cuu duong than cong . com

cuu duong than cong . com



■ Thuật ngữ:

■ Interface:

- Implementation vs. type.
- Sub-class vs. Sub-type.

■ Compile-time vs. run-time.

■ Hard-coded vs. soft-coded.

■ Composition (wrapper):

- Static reuse vs. dynamic reuse.

■ There is no free lunch!