



# CÔNG NGHỆ PHẦN MỀM

## Bài 6. Thiết kế chương trình

*ThS. Cao Minh Thành, 2023*

## 6.1 Thiết kế chương trình là gì

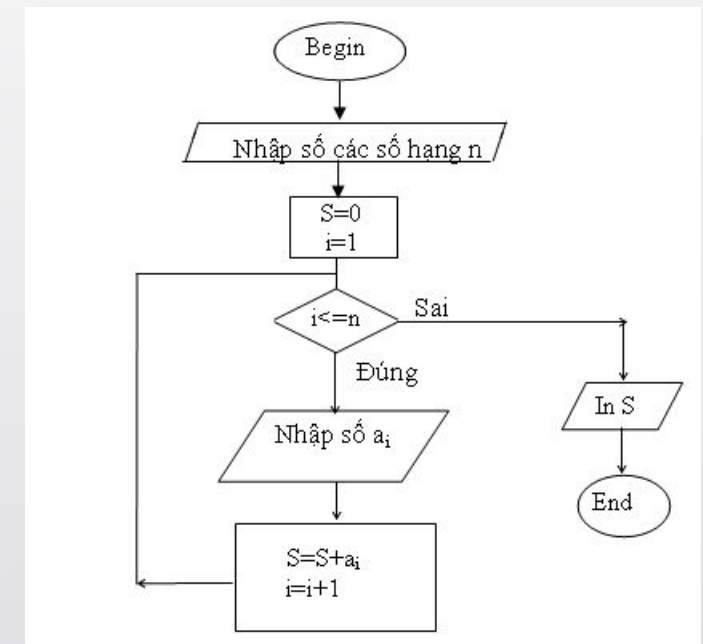
- Là thiết kế chi tiết cấu trúc bên trong của phần mềm (tính năng từng mô đun và giao diện tương ứng), gồm 2 NV cơ bản:

+ Thiết kế giải thuật

+ Thiết kế cấu trúc dữ liệu

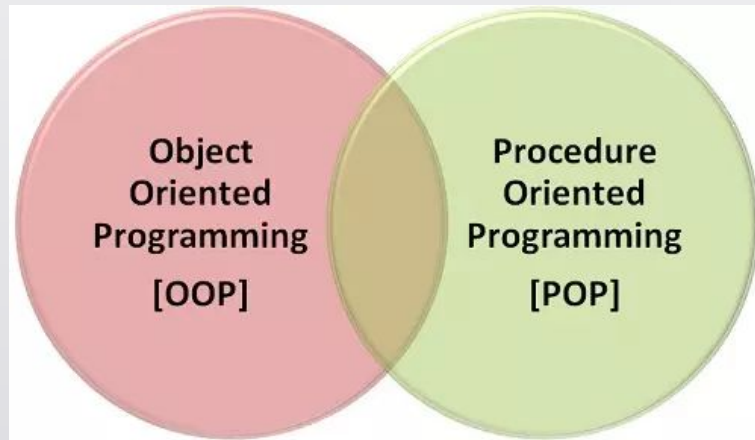
- Đầu vào là các đặc tả của giai đoạn phân tích và thiết kế HT.

- Đầu ra là các mô tả giải thuật



## 6.2 Các phương pháp thiết kế chương trình

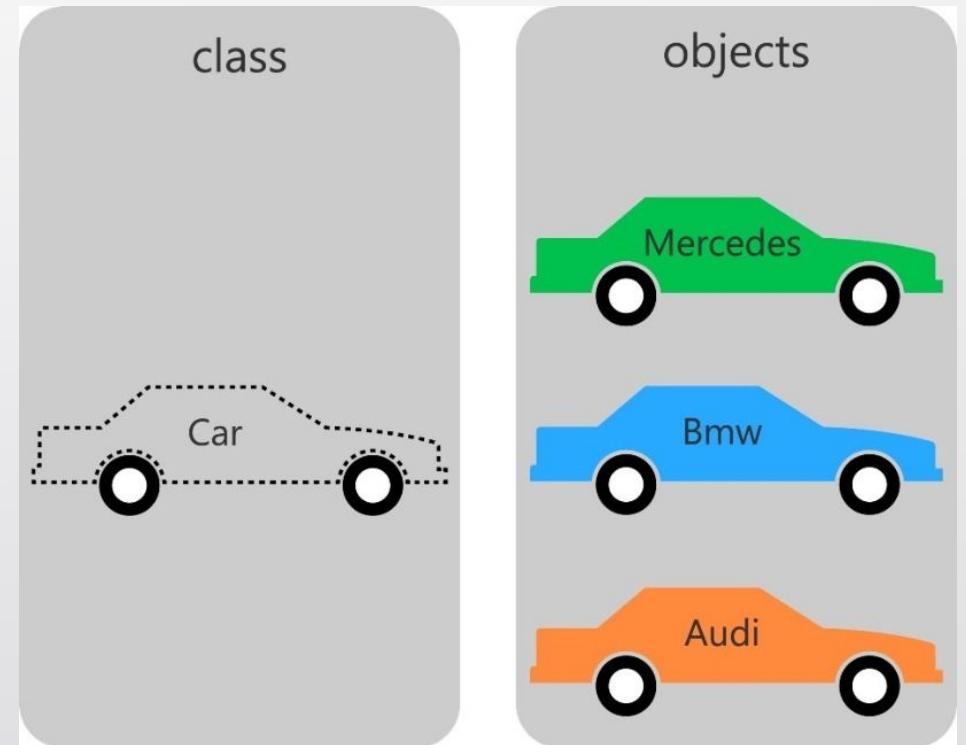
- Hướng tiến trình (process): hướng cấu trúc điều khiển
- Hướng cấu trúc dữ liệu (data)
- Hướng đối tượng



## 6.3 Hướng đối tượng

### 1. Khái niệm hướng đối tượng

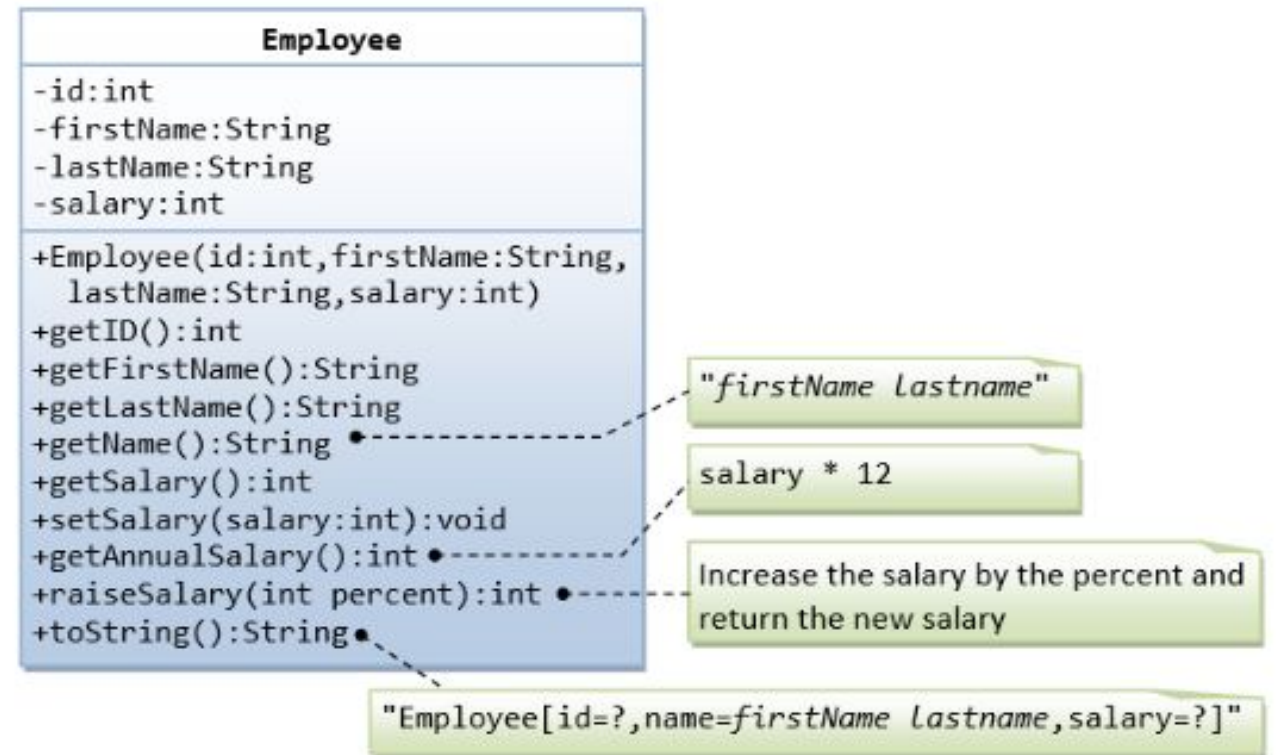
- Đối tượng là những thực thể trong hệ thống, biểu diễn các thể hiện trong thế giới thực hay là các thực thể của hệ thống.
- Lớp các thực thể là các mẫu của đối tượng và dùng để tạo các đối tượng.
- Lớp đối tượng có thể thừa kế các tính chất và các dịch vụ (services) từ các lớp khác.



## 6.3 Hướng đối tượng

### 1. Khái niệm hướng đối tượng

- Đối tượng có một trạng thái và một tập các thao tác trên trạng thái đó.





## 6.3 Hướng đối tượng

### 2. Truyền thông giữa các đối tượng

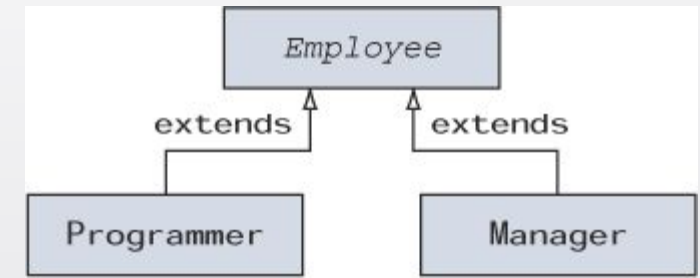
- Truyền thông điệp
- Thông điệp gồm:
  - + Tên của dịch vụ được yêu cầu bởi đối tượng gọi
  - + Bản sao các thông tin được yêu cầu để thực hiện dịch vụ và tên của đối tượng chứa kết quả của dịch vụ.
  - + Trong thực tế, thông điệp được cài đặt theo cơ chế gọi thủ tục
  - + Name = tên thủ tục
  - + Information = danh sách các tham số

```
String name = employee1.getName();
```

## 6.3 Hướng đối tượng

### 3. Khái quát hóa và kế thừa

- Các đối tượng là các thành viên của lớp định nghĩa cho kiểu các thuộc tính và các phép toán.
- Các lớp có thể sắp xếp theo kiến trúc phân cấp, có 1 lớp cao nhất là super-class. Đó là lớp khái quát nhất của 1 hay nhiều các lớp con (sub-class).
- Một lớp con kế thừa các thuộc tính và các thao tác của lớp cha và có thể thêm các thuộc tính và các phương thức mới của riêng mình.
- Sự khái quát hóa trong UML được cài đặt như kế thừa trong ngôn ngữ lập trình HĐT.



## 6.3 Hướng đối tượng

### 4. Bao gói (Encapsulation)

- Trong suốt với người dùng:

+ Người dùng không quan tâm tới cấu trúc bên trong của một đối tượng. Sự thay đổi bên trong đối tượng không ảnh hưởng đến người dùng.

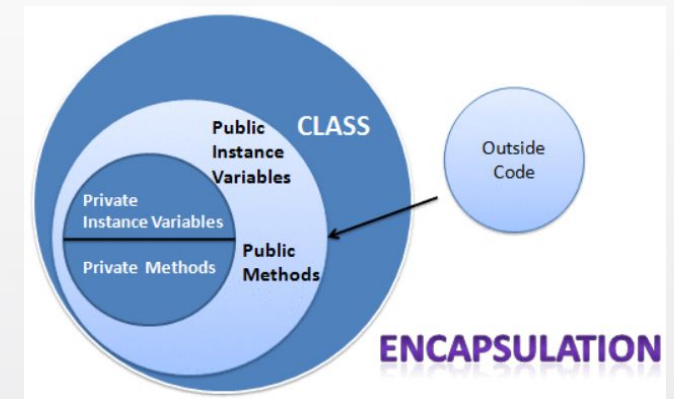
+ Mọi tác động đến đối tượng đều thông qua cơ chế truyền thông điệp.

- Cài đặt ẩn:

+ Cho phép khách truy cập và sử dụng các mục tương ứng

+ Một số lớp ẩn và không nhất thiết khách phải biết và truy nhập tới

+ Cho phép nhà thiết kế thay đổi, định nghĩa lại mà không ảnh hưởng đến ứng dụng.



```
9  class Employee
10  {
11      private string name;
12      0 references
13      public string Name
14      {
15          get { return name; }
16          set { name = value; }
17      }
18      //Auto Properties
19      0 references
20      public string Address
21      {
22          get; set;
23      }
24  }
```

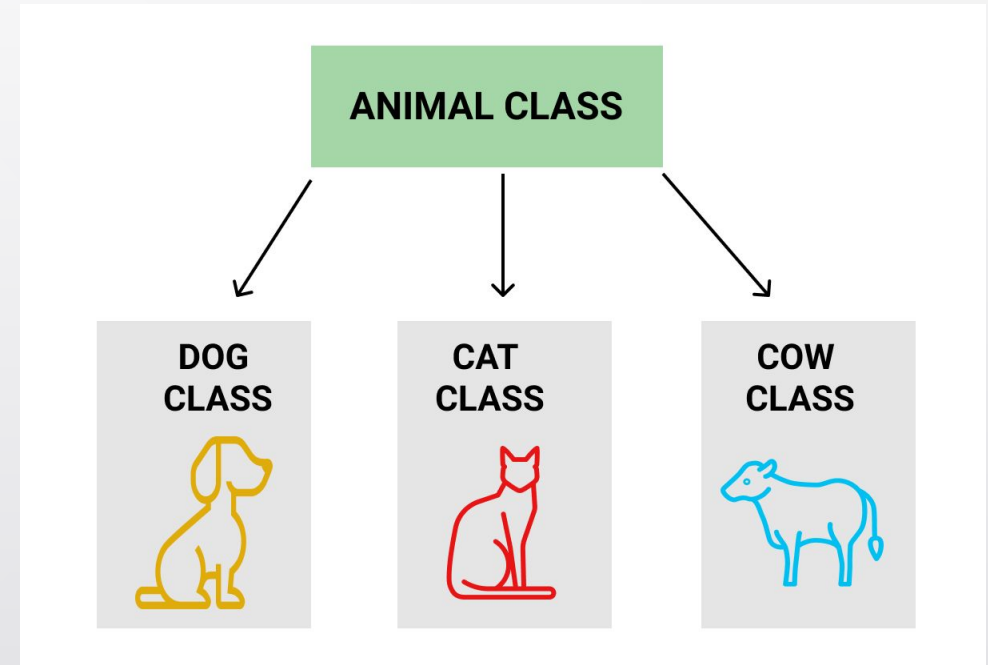


## 6.3 Hướng đối tượng

### 5. Kế thừa

- Tái sử dụng trong OPP:

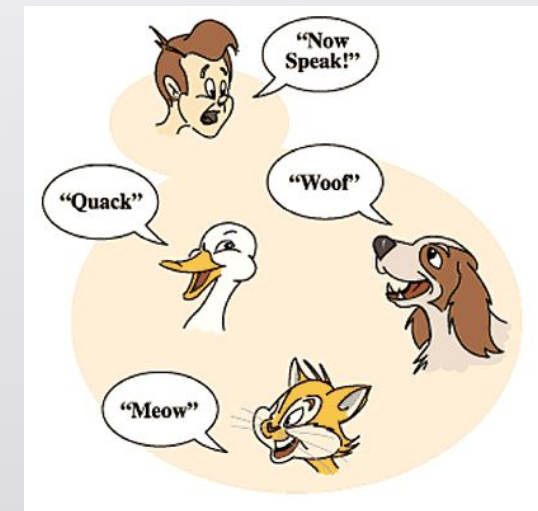
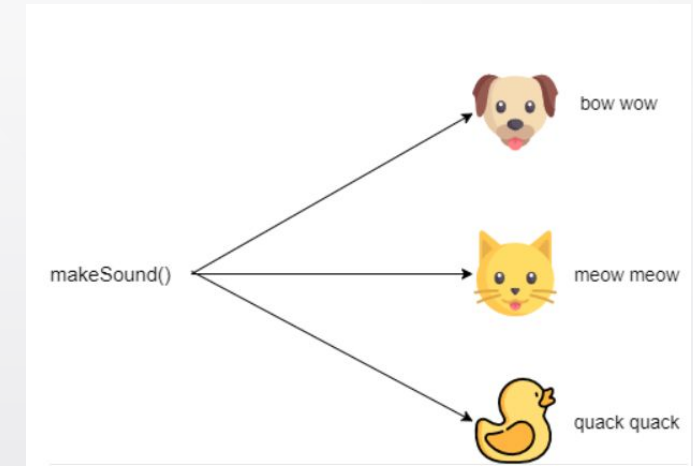
- + Tái sử dụng các lớp sẽ giảm chi phí.
- + Cách tốt nhất để tái sử dụng một lớp: tái sử dụng trực tiếp như là các trường hợp riêng.
- + Thứ 2 là dùng các biến đổi tượng lớp để xây dựng thành lớp mới
- + Thứ 3 là thể hiện nguyên tắc kế thừa
- + Các lớp cơ sở □ các lớp kế thừa



## 6.3 Hướng đối tượng

### 6. Đa hình trong OOP

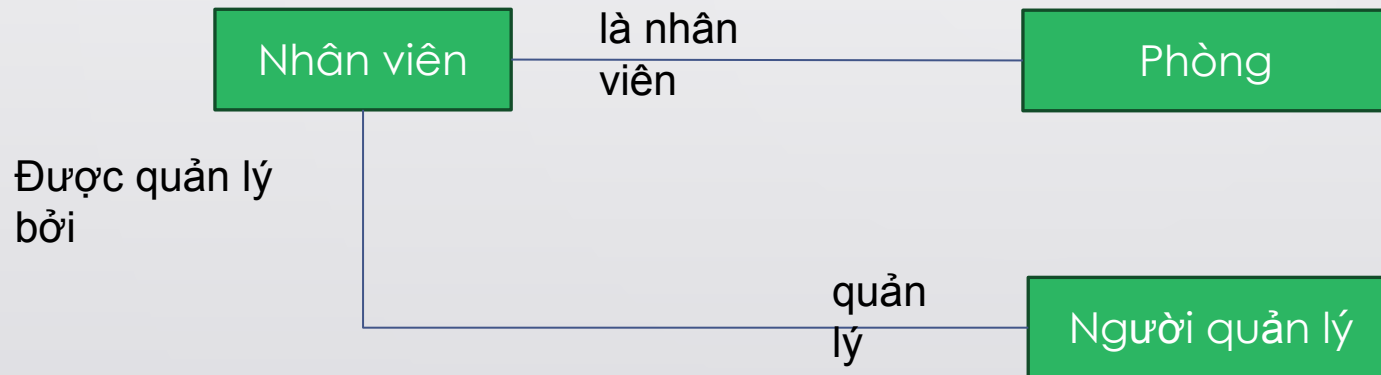
- Các lớp kế thừa định nghĩa cùng 1 hành vi (function), trình biên dịch không thể xác định được sử dụng hành vi của lớp nào.
- Có thể áp dụng các nguyên tắc sau:
  - + Chồng hàm: các tham số chỉ khác nhau về kiểu hay số lượng
  - + Chồng toán tử: định nghĩa lại các toán tử trong 1 ngôn ngữ với các dữ liệu, đối tượng mới trong các lớp cần phải xây dựng.




## 6.3 Hướng đối tượng

### 7. Các liên kết trong UML

- Các đối tượng và các lớp đối tượng tham gia vào quan hệ với các đối tượng và các lớp đối tượng khác.






## 6.3 Hướng đối tượng

### 8. Quy trình thiết kế HĐT – OOD

- Xác định ngữ cảnh và cách thức sử dụng hệ thống
- Thiết kế kiến trúc hệ thống
- Xác định các đối tượng chính của HT
- Phát triển mô hình thiết kế
- Đặc tả giao diện của các đối tượng



## 6.3 Hướng đối tượng

### 9. Đặc trưng của OOD

- Các đối tượng là trừu tượng
- Các đối tượng là độc lập và bao kín trạng thái và biểu diễn các thông tin
- Chức năng của hệ thống được biểu diễn là dịch vụ
- Việc dùng chung dữ liệu bị loại bỏ, các đối tượng truyền thông với nhau theo cơ chế truyền thông điệp.
- Các đối tượng có thể phân tán và có thể thực hiện 1 cách tuần tự hay song song



## 6.3 Hướng đối tượng

### 10. Ưu điểm của OOD

- Dễ bảo trì. Các đối tượng được hiểu như là các thực thể đứng một mình.
- Đối tượng có các thành phần tái sử dụng
- Với một HT, có thể ánh xạ ngay từ thực thể thực sang thực thể HT
- Phát triển HĐT:
  - + PT, TK và LT OOP có mối quan hệ song mục đích là khác nhau.
  - + OOA liên quan tới việc phát triển một mô hình đối tượng cho một lĩnh vực ứng dụng
  - + OOD liên quan đến việc phát triển một mô hình hệ thống hướng ĐT đáp ứng yêu cầu cài đặt.
  - + OOP liên quan đến việc thực hiện một OOD bằng việc sử dụng 1 ngôn ngữ lập trình HĐT



## 6.4 Phân tích HĐT



## 6.4 Phân tích HĐT

### 1. Năm (5) khung nhìn UML



## 6.4 Phân tích HĐT

### 2. Các phần tử cơ bản của UML



## 6.4 Phân tích HĐT

### 3. Sơ đồ Use-Case





## 6.4 Phân tích HĐT

### 4. Sơ đồ tương tác (Interact diagrams)



## 6.4 Phân tích HĐT

### 5. Sơ đồ cộng tác (Collaboration diagrams)



## 6.4 Phân tích HĐT

6. Sơ đồ diễn tiến (Sequence diagram)



## 6.4 Phân tích HĐT

### 7. Sơ đồ lớp (class diagram)



## 6.4 Phân tích HĐT

### 8. Sơ đồ State (State diagram)





## 6.5 Thiết kế HĐT

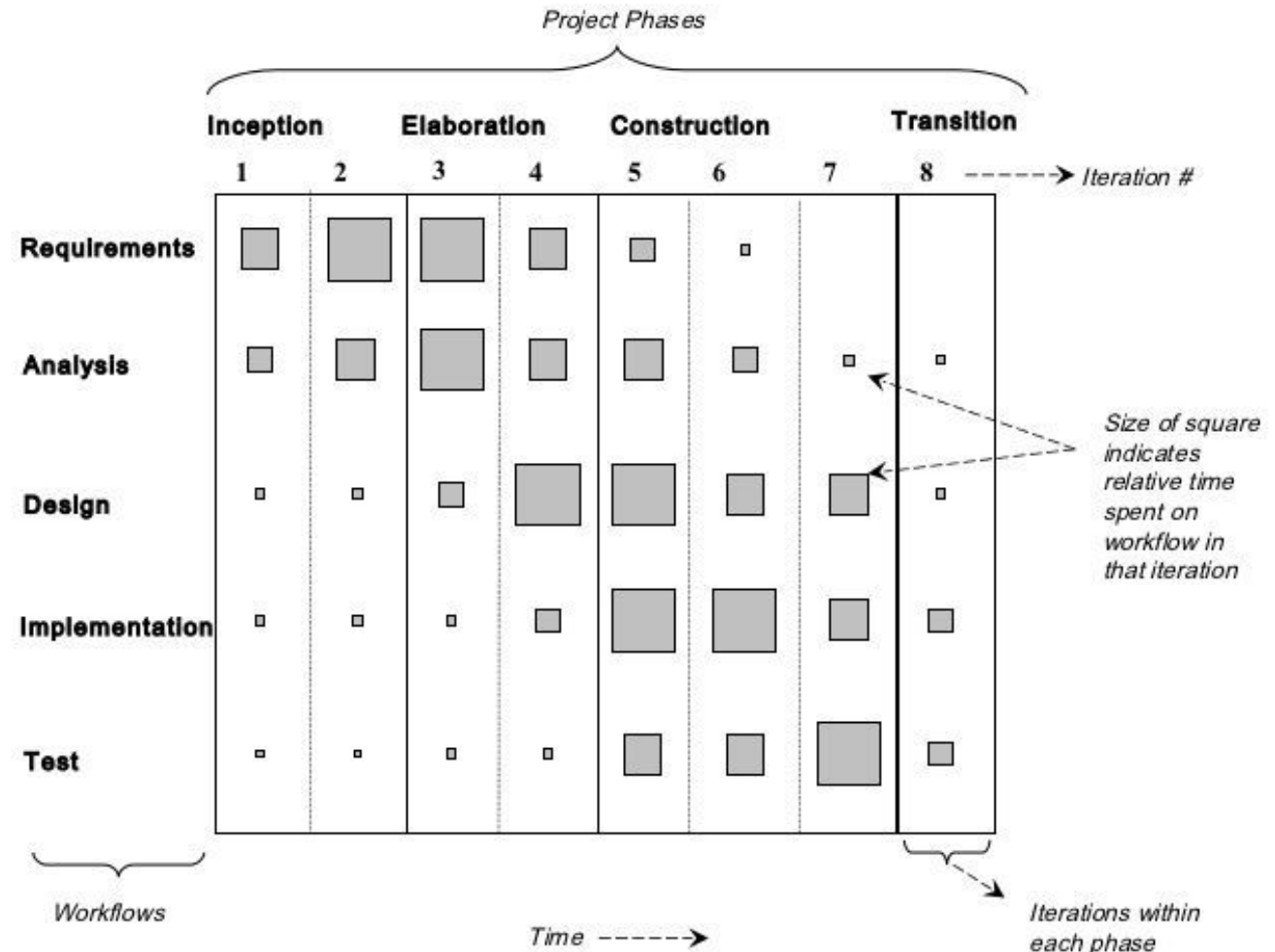
### 1. Quy trình hợp nhất – USDP process

- Là quy trình phát triển PM theo PP HĐT
- Quá trình phát triển nhằm thực hiện các yêu cầu cần phải đặc tả là:
  - + Cái gì phải làm
  - + Làm như thế nào
  - + Ai sẽ làm
- USDP được phát triển dựa trên UML và dựa theo nguyên tắc
  - + Phát triển lặp và gia tăng
  - + Phát triển HĐT – Thành phần
  - + Yêu cầu HĐT
  - + Cấu trúc hóa
  - + Kỹ thuật mô hình hóa trực quan

## 6.5 Thiết kế HĐT

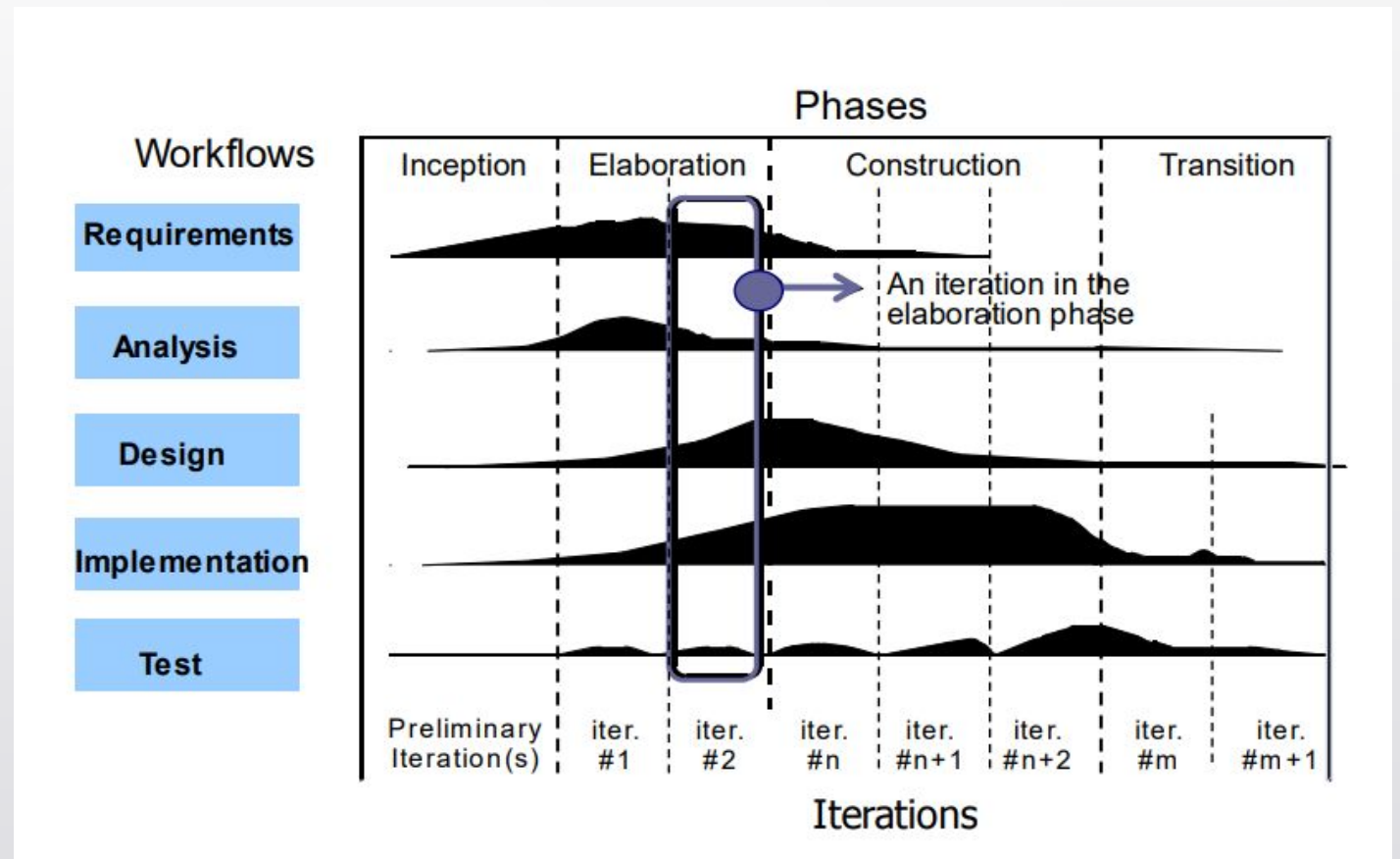
### 1. Quy trình hợp nhất – USDP process

The Unified Software Development Process



## 6.5 Thiết kế HĐT

### 1. Quy trình hợp nhất – USDP process





## 6.5 Thiết kế HĐT

### 2. Activity diagram for design process



## 6.5 Thiết kế HĐT

### 3. Thu thập và mô hình hóa yêu cầu





## 6.5 Thiết kế HĐT

### 4. Thiết kế hệ thống



## 6.5 Thiết kế HĐT

### 5. Thiết kế đối tượng và dữ liệu



## 6.6 Một số lưu ý khi mã hóa chương trình

### 1. Cấu trúc dữ liệu dễ hiểu



## 6.6 Một số lưu ý khi mã hóa chương trình

### 2. Cấu trúc thuật toán dễ hiểu