



## CHỦ ĐỀ 3

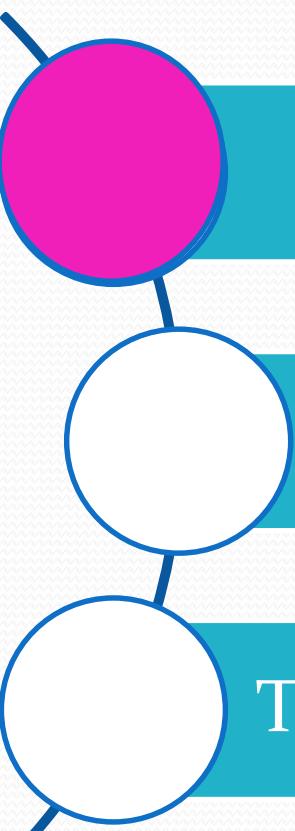


# TIẾP CẬN THEO HƯỚNG HƯỚNG ĐỐI TƯỢNG

GVGD: PHẠM THỊ KIM NGOAN

Email: ptkngoan@gmail.com

# Nội dung

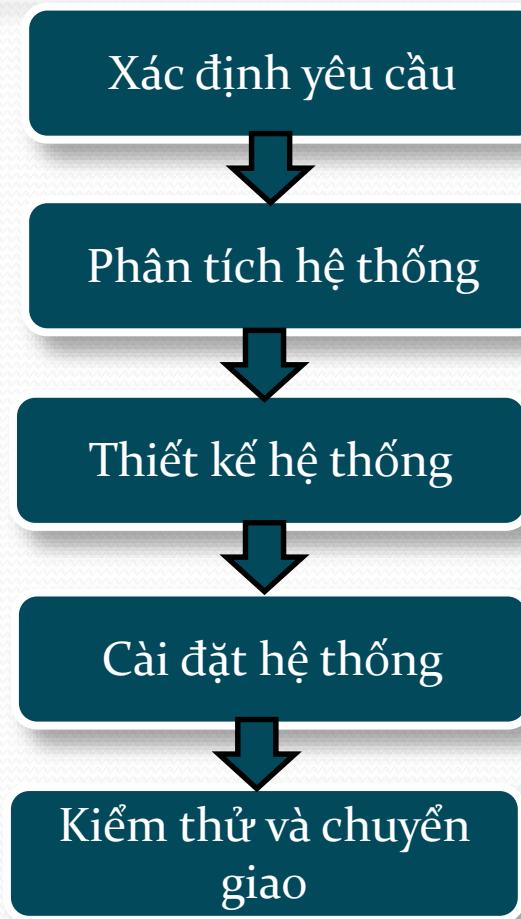


Các giai đoạn phát triển ứng dụng

Một số ký hiệu trong UML

Thiết kế chương trình theo HDT

# 1. Các giai đoạn phát triển ứng dụng



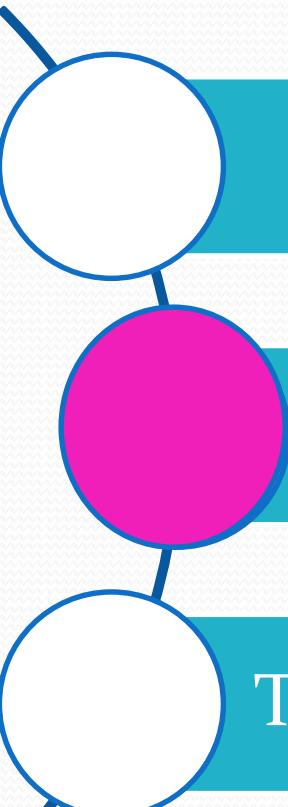
# 1. Các giai đoạn phát triển ứng dụng

---

- Xác định yêu cầu: khảo sát hệ thống => các yêu cầu chức năng và phi chức năng.
- Phân tích HT: phân tích các yêu cầu, mô hình hóa yêu cầu => sơ đồ lớp đối tượng trong hệ thống.
- Thiết kế HT: thiết kế các thuật toán, và thiết kế mô hình dữ liệu nó cho ra một mô hình lớp và dữ liệu ở mức chi tiết.
- Cài đặt HT: Các lớp của bước thiết kế sẽ được chuyển thành mã nguồn theo một ngôn ngữ lập trình theo hướng đối tượng nào đó.
- Kiểm thử và chuyển giao: test hệ thống, triển khai hệ thống.



# Nội dung



Các giai đoạn phát triển ứng dụng

Một số ký hiệu trong UML

Thiết kế chương trình theo HDT

## 2. Một số ký hiệu trong UML

### ➤ UML là gì?

- UML - Unified Modeling Language
- Là ngôn ngữ mô hình hóa hợp nhất dùng để biểu diễn hệ thống. Nói một cách đơn giản là nó dùng để tạo ra các bản vẽ nhằm mô tả thiết kế hệ thống.

### ➤ OOAD sử dụng UML bao gồm các thành phần sau:

- View (góc nhìn)
- Diagram (bản vẽ)
- Notations (ký hiệu)
- Mechanisms (qui tắc, cơ chế)

## 2. Một số ký hiệu trong UML

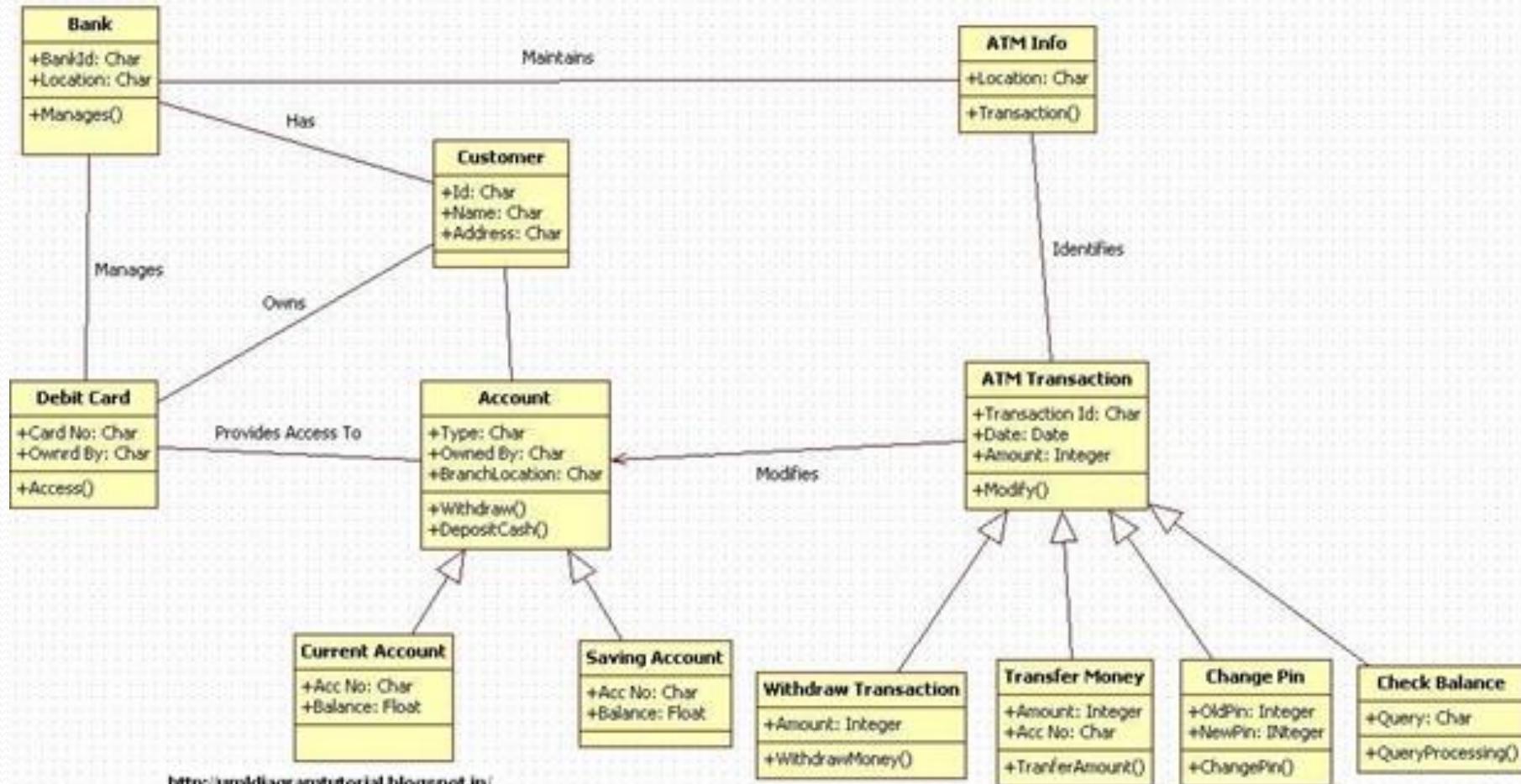
---

### ➤ Biểu đồ lớp (Class diagram – CD)

- Khung nhìn tĩnh của một hệ thống chủ yếu hỗ trợ các yêu cầu chức năng của hệ thống.
- Chỉ ra sự tồn tại của các lớp và mối quan hệ giữa chúng trong bản thiết kế logic của một hệ thống.
- Chỉ rõ đặc trưng của lớp, các thuộc tính và tác vụ chính của mỗi lớp.

## 2. Một số ký hiệu trong UML

### ➤ Biểu đồ lớp (Class diagram – CD)



## 2. Một số ký hiệu trong UML

### ➤ Ký hiệu lớp

Tên lớp
- Thuộc tính 1 - Thuộc tính 2 - ...

+ Phương thức 1() + Phương thức 2() + ...

### Ký hiệu tầm vực:

- private access
- # protected access
- + public access

## 2. Một số ký hiệu trong UML

### ➤ Ví dụ lớp

Tên lớp	
Các thuộc tính	
Các phương thức	
	<b>student</b> <ul style="list-style-type: none"> <li>- name: string</li> <li>- rollNo: int</li> <li>+ setName(): void</li> <li>+ setRollNo(): void</li> </ul>



## 2. Một số ký hiệu trong UML

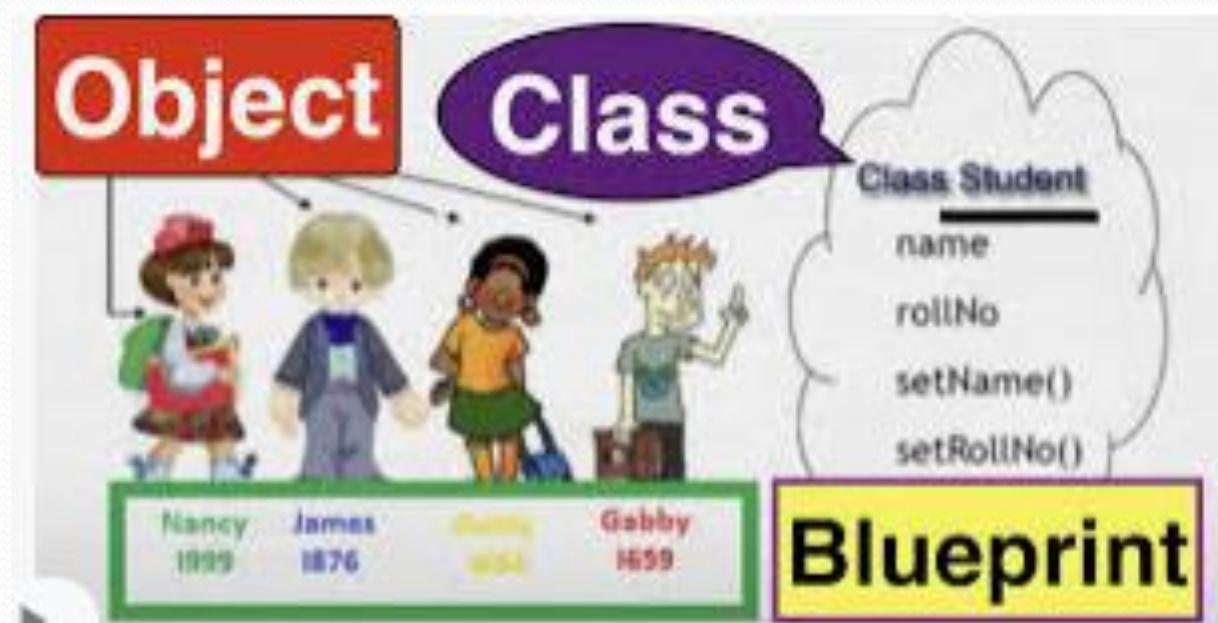
### ➤ Ví dụ đối tượng

Tên đối tượng: Tên lớp

- thuộc tính = giá trị thuộc tính

Nancy:student

- name = Nancy
- rollNo = 1999



## 2. Một số ký hiệu trong UML

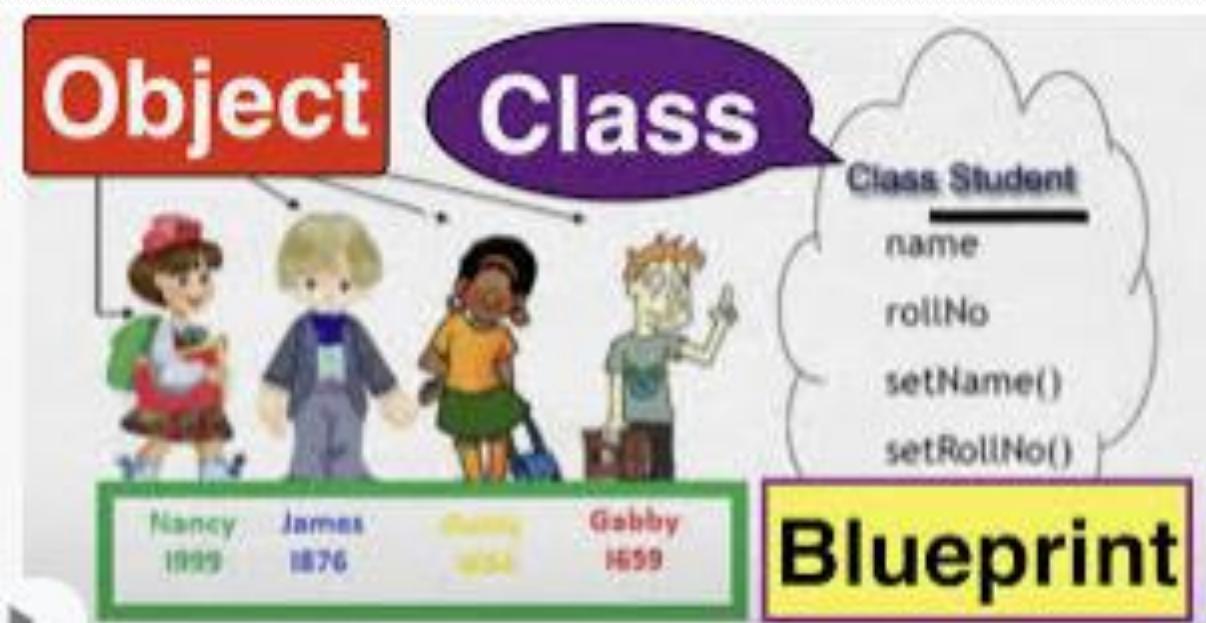
### ➤ Ví dụ đối tượng

#### Tên đối tượng

- thuộc tính = giá trị thuộc tính

#### Nancy

- name = Nancy
- rollNo = 1999



## 2. Một số ký hiệu trong UML

### ➤ Biểu diễn thuộc tính

- Chỉ ra tên, kiểu và giá trị mặc định nếu có
  - **attributeName : Type [= Default]**
- Tuân theo quy ước đặt tên của ngôn ngữ cài đặt và của ứng dụng.
- Kiểu (type) nên là kiểu dữ liệu cơ bản trong ngôn ngữ thực thi.

## 2. Một số ký hiệu trong UML

### ➤ Biểu diễn thuộc tính

- Ví dụ: Lớp sinh viên có các thuộc tính: mã số sinh viên, họ tên, điểm trung bình
- Biểu diễn các thuộc tính của sinh viên:
  - maSV: string
  - tenSV: string
  - diemTB: float = 0

## 2. Một số ký hiệu trong UML

### ➤ Mô tả phương thức

- Tên phương thức:
  - Mô tả kết quả
  - Sử dụng góc nhìn của đối tượng khách (đối tượng gọi)
  - Nhứt quán giữa các lớp
- ký hiệu của phương thức:  
**+ perationName(parameters): returnType**

parameter: in|out|inout parameterName:Type[= Default]

## 2. Một số ký hiệu trong UML

### ➤ Mô tả phương thức

- Ví dụ: Lớp sinh viên có các phương thức: Xuất thông tin sinh viên, xét xem sinh viên có được học bổng hay không.
- Biểu diễn các phương thức của sinh viên:
  - + Xuat(): void hoặc + void Xuat()
  - + NhanHocBong(in float s): int hoặc
  - + int NhanHocBong(float d)

## 2. Một số ký hiệu trong UML

### ➤ Thành viên tĩnh

- Lớp sinh viên có các thuộc tính: mã số sinh viên, họ tên, tổng số tính chỉ (thành viên tĩnh); các phương thức: Xuất thông tin sinh viên, xét xem sinh viên có được nhận học bổng hay không.

SinhVien
- maSV: string
- tenSV: string
- <u>tongTC</u> : int
+ void Xuat()
+ int NhanHocBong(float f)

## 2. Một số ký hiệu trong UML

---

### ➤ Quan hệ giữa các lớp

- Quan hệ kết hợp (Association)
- Quan hệ kết nhập (Aggregation) và hợp thành (Composition)
- Quan hệ phụ thuộc (Dependency)
- Quan hệ khái quát hóa (Generalization)
- Quan hệ hiện thực hóa (Realization)

## 2. Một số ký hiệu trong UML

### ➤ Quan hệ kết hợp (Association)

- Mỗi liên hệ ngữ nghĩa giữa hai hay nhiều lớp chỉ ra sự liên kết giữa các thể hiện của chúng.
- Mỗi quan hệ về mặt cấu trúc chỉ ra các đối tượng của lớp này có kết nối với các đối tượng của lớp khác.
- Ký hiệu



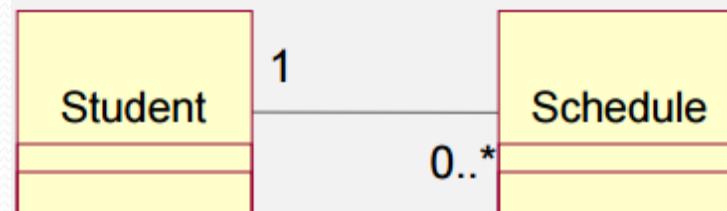
## 2. Một số ký hiệu trong UML

### ➤ Quan hệ kết hợp (Association)

- Chiều của mối quan hệ



- Bội số của mối quan hệ
  - 1: một và chỉ một
  - 0..\* hoặc \*: 0 đến nhiều
  - 1..\*: một đến nhiều
  - 0..1: không hay một
  - m..n: từ m đến n ( $m, n \in \mathbb{N}$ )
  - Mặc định là 1



## 2. Một số ký hiệu trong UML

### ➤ Quan hệ kết tập (Aggregation)

- Kết tập là mối quan hệ “là một phần” (“is a part-of”)
- Ký hiệu:



## 2. Một số ký hiệu trong UML

### ➤ Quan hệ hợp thành (Composition)

- Là quan hệ Association và khi một đối tượng b thuộc classB bị hủy thì đối tượng classA bị hủy theo.
- Ký hiệu:



## 2. Một số ký hiệu trong UML

### ➤ Quan hệ phụ thuộc (Dependency)

- Quan hệ 1 chiều: 1 lớp phụ thuộc vào lớp khác (bị sử dụng)
- Ký hiệu:



## 2. Một số ký hiệu trong UML

### ➤ Quan hệ tổng quát hóa (Generalization)

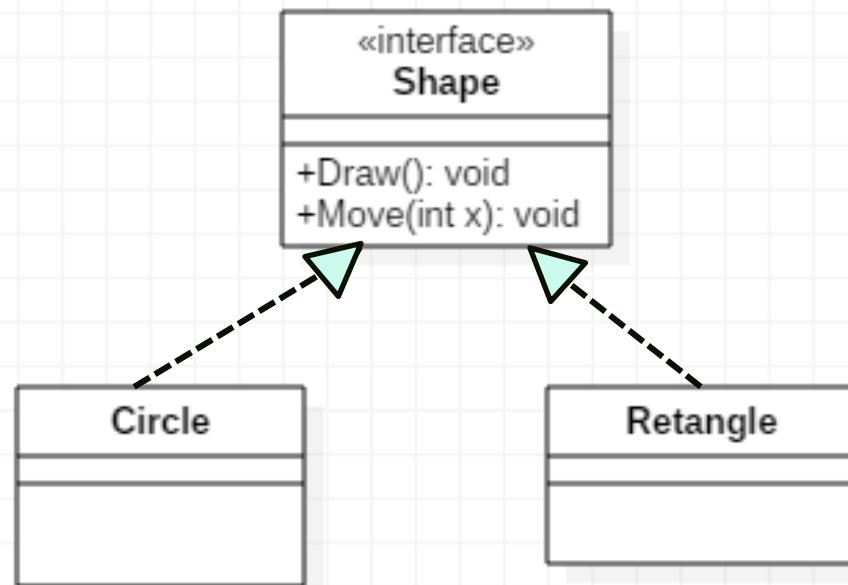
- Mỗi quan hệ giữa các lớp trong đó một lớp chia sẻ cấu trúc và/hoặc hành vi với một hoặc nhiều lớp khác.
- Xác định sự phân cấp về mức độ trừu tượng hóa trong đó lớp con kế thừa từ một hoặc nhiều lớp cha
- Ký hiệu:



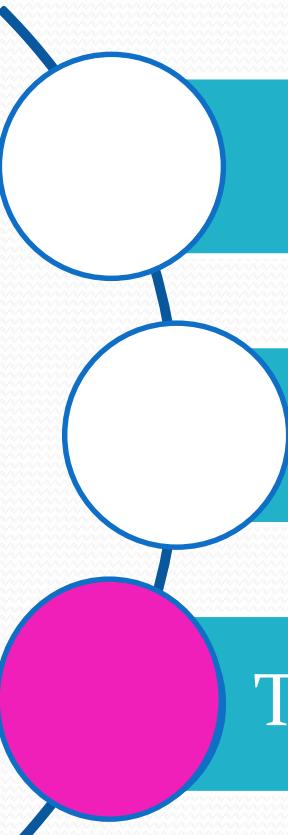
## 2. Một số ký hiệu trong UML

### ➤ Quan hệ hiện thực hóa (Realization)

- Mỗi quan hệ ngữ nghĩa giữa giao diện và lớp để thực hiện cài đặt các dịch vụ đã được khai bao trong các giao diện (Interface)



# Nội dung



Các giai đoạn phát triển ứng dụng

Một số ký hiệu trong UML

Thiết kế chương trình theo HDT

### 3. Thiết kế chương trình theo HĐT

---

#### ➤ Các bước thiết kế chương trình

- Thiết kế các lớp đối tượng (class), mối quan hệ giữa chúng.
- Thiết kế chi tiết các phương thức cho lớp đối tượng.
- Thiết kế chương trình chính.

# Kết thúc chủ đề 3

---

