

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  
ĐỘC LẬP – TỰ DO – HẠNH PHÚC



## GIÁO TRÌNH

# LẬP TRÌNH JAVA CĂN BẢN

## MODULE: JAVA CORE

### LAB 7: HƯỚNG ĐỐI TƯỢNG OBJECT-ORIENTED PROGRAMMING OOP

Dương Nguyễn Phú Cường

Năm 2024

<https://nentang.vn>

# MỤC LỤC

MỤC LỤC .....	I
DANH MỤC HÌNH .....	III
DANH MỤC BẢNG .....	IV
DANH MỤC TỪ VIẾT TẮT .....	V
Chương 1: CLASS, OBJECTS, PACKAGES.....	1
Bài 1: Thiết kế class Circle (100đ).....	1
Mục 1: Thông tin.....	1
Mục 2: Yêu cầu .....	1
Bài 2: Thiết kế class Rectangle (100đ) .....	3
Mục 1: Thông tin.....	3
Mục 2: Yêu cầu .....	3
Bài 3: Thiết kế class Employee (100đ) .....	5
Mục 1: Thông tin.....	5
Mục 2: Yêu cầu .....	5
Bài 4: Thiết kế class Processor và MathProcessor (100đ) .....	6
Mục 1: Thông tin.....	6
Mục 2: Yêu cầu .....	6
Bài 5: Thiết kế class BankAccount (100đ) .....	7
Mục 1: Thông tin.....	7
Mục 2: Yêu cầu .....	7
Bài 6: Viết chương trình quản lý Nhân viên (100đ) .....	9
Mục 1: Thông tin.....	9
Mục 2: Yêu cầu .....	9
Bài 7: Viết chương trình quản lý Tài khoản ngân hàng (100đ) .....	9
Mục 1: Thông tin.....	9
Mục 2: Yêu cầu .....	9

Chương 2: KẾ THỪA (INHERITANCE), TRỪU TƯỢNG (ABSTRACT), HỢP ĐỒNG (INTERFACE).....	11
Bài 1: Thiết kế class ConNguoi, SinhVien, GiangVien (100đ) .....	11
Mục 1: Thông tin.....	11
Mục 2: Yêu cầu.....	11
Bài 2: Thiết kế interface IBankService (100đ).....	12
Mục 1: Thông tin.....	12
Mục 2: Yêu cầu.....	12
XIN CẢM ƠN ! .....	B

## DANH MỤC HÌNH

Hình 1.1: UML cLass-diagram Circle .....	1
Hình 1.2: UML class-diagram Rectangle .....	3
Hình 1.3: UML class-diagram Employee.....	5
Hình 1.4: UML class-diagram BankAccount .....	7
Hình 2.1: UML class-diagram ConNguoi, SinhVien và GiangVien .....	11
Hình 2.2: UML class-diagram IBankService và Vietcombank, Agribank	12

# DANH MỤC BẢNG

Bảng 1.1: Kịch bản test case BankAccount .....	8
--	---

## DANH MỤC TỪ VIẾT TẮT

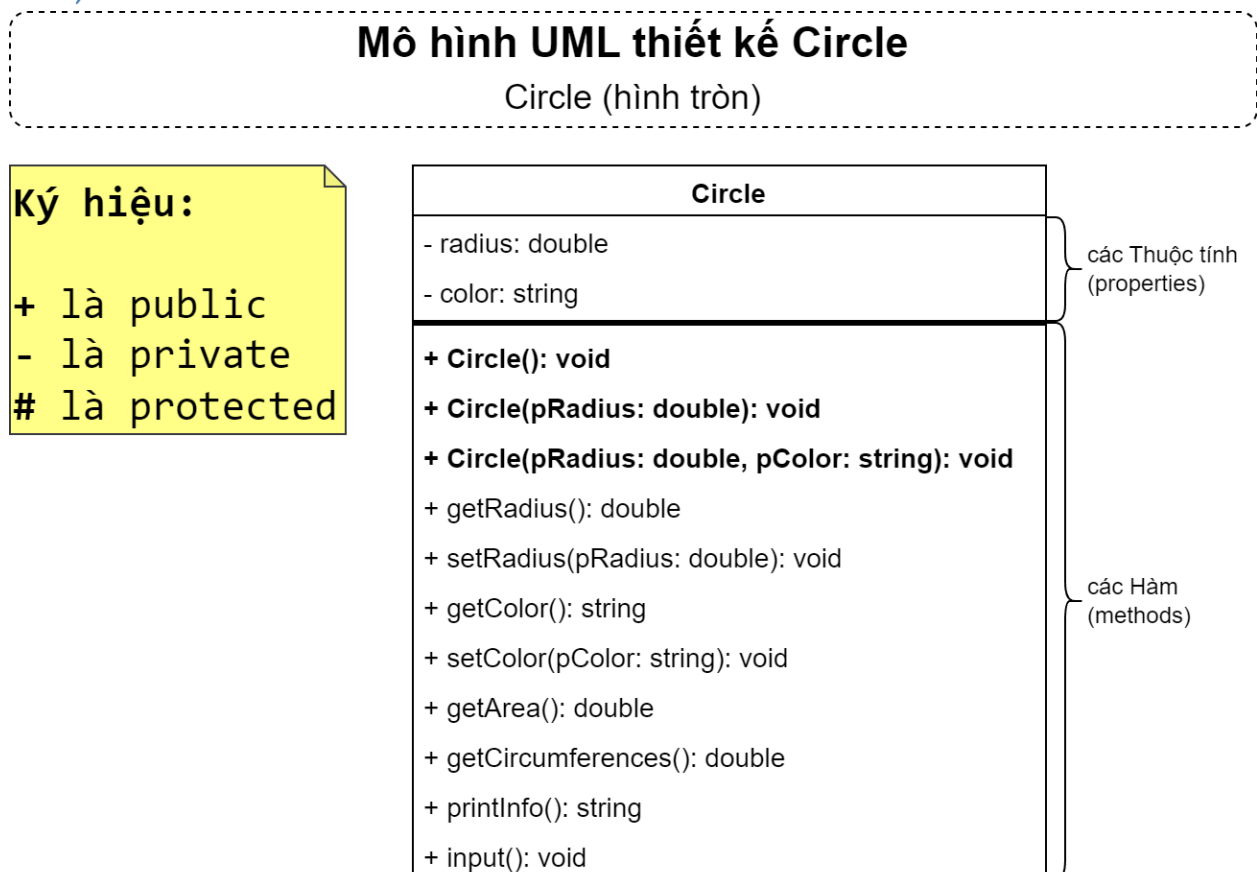
STT	Từ viết tắt	Tiếng Việt	Tiếng Anh
1.	CSDL	Cở sở dữ liệu	Database
2.			
3.			
4.			
5.			

# Chương 1: CLASS, OBJECTS, PACKAGES

## Bài 1: Thiết kế class Circle (100đ)

### Mục 1: Thông tin

#### a) Sơ đồ UML Class



Hình 1.1: UML cLass-diagram Circle

#### b) Công thức

- Công thức tính diện tích hình tròn:  
$$\text{diện tích hình tròn} = \pi * \text{bán kính}^2$$
- Công thức tính chu vi hình tròn:  
$$\text{chu vi hình tròn} = \pi * \text{bán kính} * 2$$

### Mục 2: Yêu cầu

- Xây dựng class theo sơ đồ UML đã thiết kế: (60đ)
  - Các thuộc tính bán kính (radius) và màu sắc (color) có thuộc tính truy xuất (access modifier) là private.
  - Có 3 hàm khởi tạo (constructor):

- Hàm khởi tạo không có tham số (mặc định): thì khởi tạo giá trị mặc định là: `radius=1.0` và `color="red"`.
- Hàm khởi tạo 1 tham số: nhận vào tham số `pRadius`, gán giá trị tham số này vào thuộc tính `radius`.
- Hàm khởi tạo 2 tham số: nhận vào tham số `pRadius` và `pColor`, gán các giá trị vào các thuộc tính tương ứng.
- Các hàm `get***()` dùng để trả về giá trị các thuộc tính tương ứng.
- Các hàm `set***()` dùng để gán giá trị các thuộc tính tương ứng.
- Hàm `getArea()` dùng để tính diện tích.
- Hàm `getCircumferences()` dùng để tính chu vi.
- Hàm `printInfo()` dùng để in thông tin `radius` và `color` ra màn hình.
- Hàm `input()` dùng để thu thập thông tin từ màn hình nhập liệu:
  - Cho phép người dùng nhập thông tin `radius` và `color`.
- b. Thực hiện khởi tạo đối tượng (object) từ class vừa thiết kế và thực hiện gọi sử dụng các hàm từ đối tượng đó. (40đ)

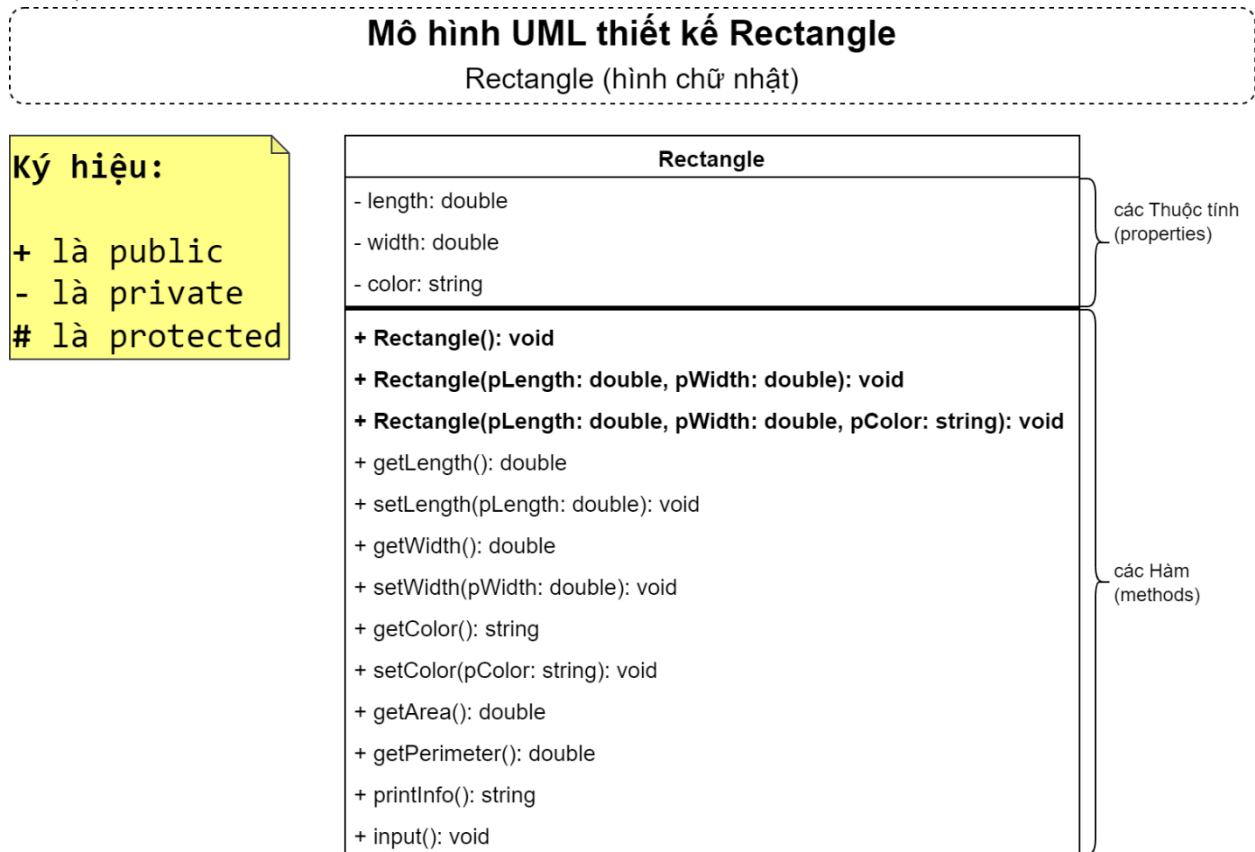
-----📖-----



## Bài 2: Thiết kế class Rectangle (100đ)

### Mục 1: Thông tin

#### a) Sơ đồ UML Class



Hình 1.2: UML class-diagram Rectangle

#### b) Công thức

- Công thức tính diện tích hình chữ nhật:

$$\text{diện tích hcn} = \text{chiều dài} * \text{chiều rộng}$$

- Công thức tính chu vi hình chữ nhật:

$$\text{chu vi hcn} = \frac{(\text{chiều dài} + \text{chiều rộng})}{2}$$

### Mục 2: Yêu cầu

- Xây dựng class theo sơ đồ UML đã thiết kế: (60đ)
  - Các thuộc tính chiều dài (length), chiều rộng (width) và màu sắc (color) có thuộc tính truy xuất (access modifier) là private.
  - Có 3 hàm khởi tạo (constructor):
    - Hàm khởi tạo không có tham số (mặc định): thì khởi tạo giá trị mặc định là: length=1.0, width=1.0 và color="red".

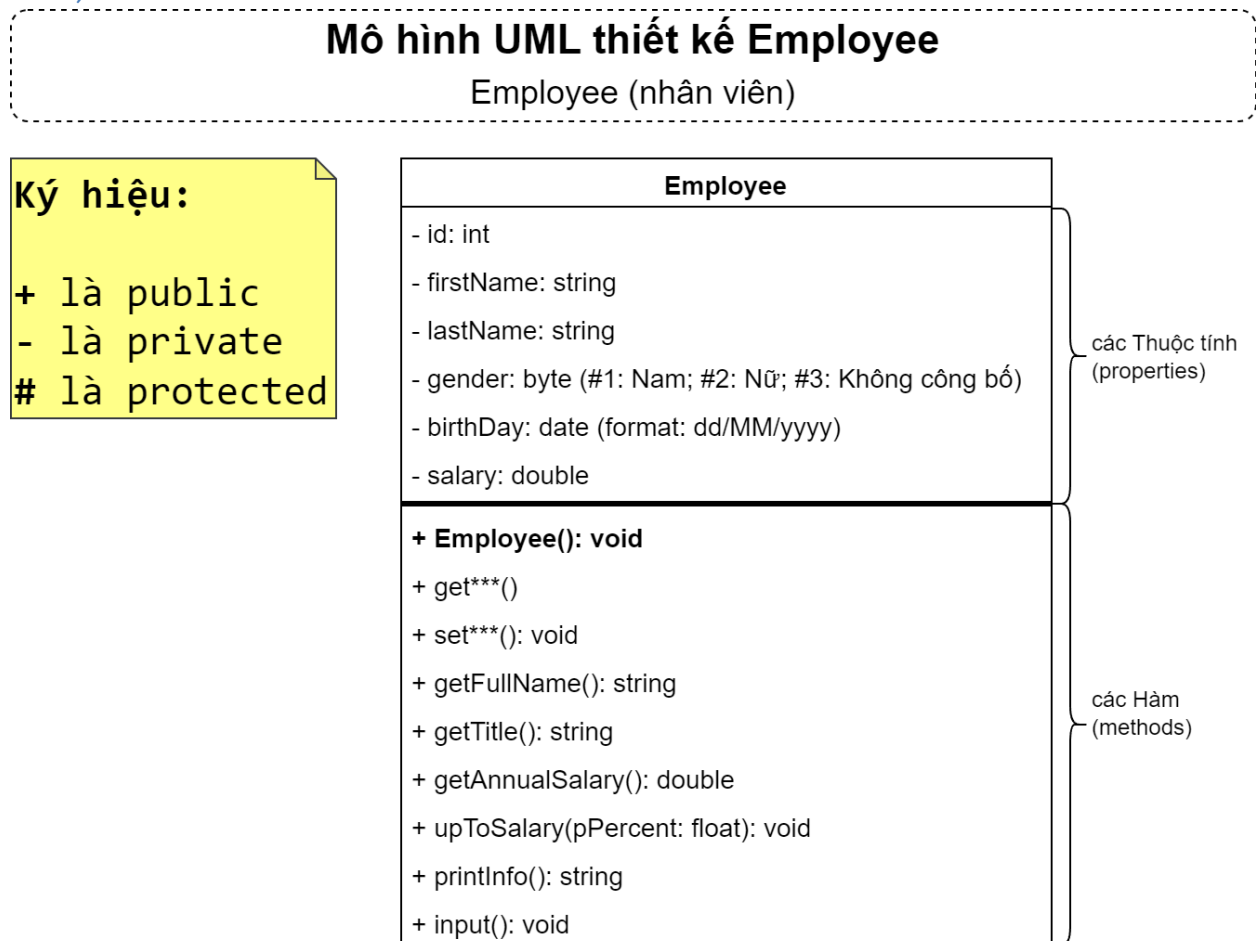
- Hàm khởi tạo 2 tham số: nhận vào tham số pLength, pWidth, gán giá trị tham số này vào thuộc tính length, width tương ứng.
  - Hàm khởi tạo 3 tham số: nhận vào tham số pLength, pWidth và pColor, gán các giá trị vào các thuộc tính tương ứng.
  - Các hàm `get***()` dùng để trả về giá trị các thuộc tính tương ứng.
  - Các hàm `set***()` dùng để gán giá trị các thuộc tính tương ứng.
  - Hàm `getArea()` dùng để tính diện tích.
  - Hàm `getPerimeter()` dùng để tính chu vi.
  - Hàm `printInfo()` dùng để in thông tin ra màn hình.
  - Hàm `input()` dùng để thu thập thông tin từ màn hình nhập liệu:
    - Cho phép người dùng nhập thông tin length, width và color.
- b. Thực hiện khởi tạo đối tượng (object) từ class vừa thiết kế và thực hiện gọi sử dụng các hàm từ đối tượng đó. (40đ)

-----&📖&-----

## Bài 3: Thiết kế class Employee (100đ)

### Mục 1: Thông tin

#### a) Sơ đồ UML Class



Hình 1.3: UML class-diagram Employee

### Mục 2: Yêu cầu

- a. Xây dựng class theo sơ đồ UML đã thiết kế: (60đ)
  - Các thuộc tính `id`, `firstName` (tên), `lastName` (họ + tên lót), `gender` (giới tính), `birthDay` (ngày sinh), `salary` (lương) có thuộc tính truy xuất (access modifier) là `private`.
  - Có các hàm khởi tạo (constructor):
    - Hàm khởi tạo không có tham số (mặc định).
  - Các hàm `get***()` dùng để trả về giá trị các thuộc tính tương ứng.
  - Các hàm `set***()` dùng để gán giá trị các thuộc tính tương ứng.
  - Hàm `getFullName()` dùng để trả về tên đầy đủ (bao gồm họ + tên lót + tên).

- Hàm `getTitle()` dùng để trả về danh xưng trong tiếng Anh với các điều kiện sau:
  - Nếu giới tính là Nữ:
    - Nếu tuổi  $\leq 18$ : gọi là Miss.
    - Nếu tuổi  $> 18$ : gọi là Ms.
  - Nếu giới tính là Nam:
    - Nếu tuổi  $\leq 35$ : gọi là Mr.
    - Nếu tuổi  $> 35$ : gọi là Sir.
- Hàm `getAnnualSalary()` dùng để tính tổng lương hàng năm:  $\text{salary} * 12$  tháng.
- Hàm `upToSalary(pPercent)` dùng để tăng lương theo phần trăm:  $\text{salary} * p\text{Percent}$ .
- Hàm `printInfo()` dùng để in thông tin ra màn hình.
- Hàm `input()` dùng để thu thập thông tin từ màn hình nhập liệu.
- b. Thực hiện khởi tạo đối tượng (object) từ class vừa thiết kế và thực hiện gọi sử dụng các hàm từ đối tượng đó. (40đ)

-----📖-----

## Bài 4: Thiết kế class `Processor` và `MathProcessor` (100đ)

### Mục 1: Thông tin

#### a) Sơ đồ UML Class

### Mục 2: Yêu cầu

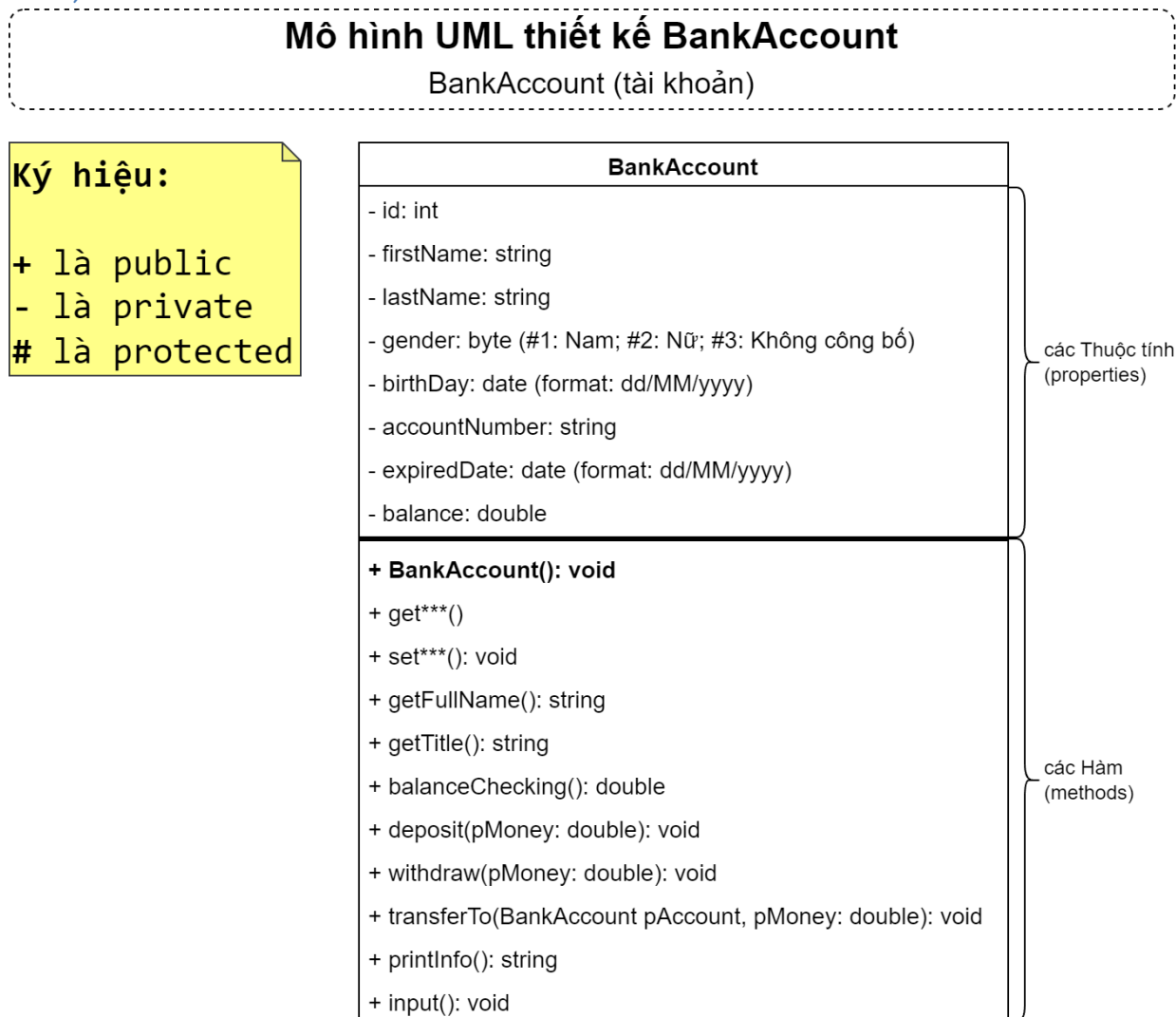
- a. Xây dựng class theo sơ đồ UML đã thiết kế: (60đ)

-

## Bài 5: Thiết kế class BankAccount (100đ)

### Mục 1: Thông tin

#### b) Sơ đồ UML Class



Hình 1.4: UML class-diagram BankAccount

### Mục 2: Yêu cầu

b. Xây dựng class theo sơ đồ UML đã thiết kế: (60đ)

- Các thuộc tính `id`, `firstName` (tên), `lastName` (họ + tên lót), `gender` (giới tính), `birthDay` (ngày sinh), `accountNumber` (số tài khoản), `expiredDate` (ngày hết hạn thẻ), `balance` (số dư tài khoản) có thuộc tính truy xuất (access modifier) là `private`.
- Có các hàm khởi tạo (constructor):
  - Hàm khởi tạo không có tham số (mặc định).
- Các hàm `get***()` dùng để trả về giá trị các thuộc tính tương ứng.
- Các hàm `set***()` dùng để gán giá trị các thuộc tính tương ứng.

- Hàm `getFullName()` dùng để trả về tên đầy đủ (bao gồm họ + tên lót + tên).
- Hàm `getTitle()` dùng để trả về danh xưng trong tiếng Anh với các điều kiện sau:
  - Nếu giới tính là Nữ:
    - Nếu tuổi  $\leq 18$ : gọi là Miss.
    - Nếu tuổi  $> 18$ : gọi là Ms.
  - Nếu giới tính là Nam:
    - Nếu tuổi  $\leq 35$ : gọi là Mr.
    - Nếu tuổi  $> 35$ : gọi là Sir.
- Hàm `balanceChecking()` dùng để kiểm tra số dư trong tài khoản.
- Hàm `deposit(pMoney)` dùng để nạp tiền vào tài khoản.
- Hàm `withdraw(pMoney)` dùng để rút tiền khỏi tài khoản.
  - Điều kiện: không được rút tiền lớn hơn số dư tài khoản.
- Hàm `transferTo(pAccount, pMoney)` dùng để chuyển số tiền từ tài khoản hiện tại sang tài khoản được chỉ định.
  - Điều kiện: không được chuyển tiền lớn hơn số dư tài khoản.
- Hàm `printInfo()` dùng để in thông tin ra màn hình.
- Hàm `input()` dùng để thu thập thông tin từ màn hình nhập liệu.
- c. Thực hiện khởi tạo đối tượng (object) từ class vừa thiết kế và thực hiện gọi sử dụng các hàm từ đối tượng đó. (30đ)
- d. Thực hiện test case theo kịch bản sau: (10đ)

Tài khoản A	Tài khoản B
id: 1 firstName: Cuong lastName: Duong Nguyen Phu gender: 1 birthDay: 11/06/1989 accountNumber: 11100089223 expiredDate: 31/12/2026 balance: 12,000,000đ	id: 2 firstName: B lastName: Nguyen Thi gender: 2 birthDay: 27/08/2001 accountNumber: 22200078201 expiredDate: 31/12/2024 balance: 3,500,000đ
Nạp tiền lần 1: +3,000,000đ	Nạp tiền lần 1: +200,000đ
Nạp tiền lần 2: +2,000,000đ	Rút tiền lần 1: -500,000đ
Rút tiền lần 1: -700,000đ	Chuyển tiền cho A lần 1: -200,000đ
Nhận tiền từ B: +200,000đ	
Chuyển cho B lần 1: -1,500,000đ	
	Nhận tiền từ A: +1,500,000đ
Kiểm tra tài khoản: 15,000,000đ	Kiểm tra tài khoản: 4,500,000đ

Bảng 1.1: Kịch bản test case BankAccount



## Bài 6: Viết chương trình quản lý Nhân viên (100đ)

### Mục 1: Thông tin

Sử dụng class Employee đã thiết kế ở bài 3.

### Mục 2: Yêu cầu

a. Tạo menu chương trình, in ra với format như sau: (10đ)

```
===== MENU CHUONG TRINH =====
+ 1. Them moi nhan vien.           +
+ 2. Xem danh sach nhan vien.      +
+ 3. Tim kiem nhan vien theo id.    +
+ 4. Xoa nhan vien.                 +
+ 5. Cap nhat nhan vien.            +
+ 0. Thoat chuong trinh.           +
=====
Moi ban chon: _
```

b. Cho người dùng tùy chọn menu sử dụng. Tương ứng với từng menu người dùng đã chọn, chương trình thực hiện chức năng tương ứng. (90đ)

-----📖-----

## Bài 7: Viết chương trình quản lý Tài khoản ngân hàng (100đ)

### Mục 1: Thông tin

Sử dụng class BankAccount đã thiết kế ở bài 4.

### Mục 2: Yêu cầu

a. Tạo menu chương trình, in ra với format như sau: (10đ)

```
===== MENU CHUONG TRINH =====
+ 1. Them moi tai khoan.           +
+ 2. Xem danh sach tai khoan.      +
+ 3. Tim kiem tai khoan theo id, so tai khoan. +
+ 4. Xoa tai khoan.                 +
+ 5. Cap nhat tai khoan.            +
+ 6. Su dung tai khoan.             +
+ 0. Thoat chuong trinh.           +
=====
Moi ban chon: _
```

b. Cho người dùng tùy chọn menu sử dụng. Tương ứng với từng menu người dùng đã chọn, chương trình thực hiện chức năng tương ứng. (90đ)

- Khi người dùng chọn menu 6 “su dung tai khoan”, sẽ hiển thị danh sách tài khoản và cho người dùng lựa chọn:

```
=== Danh sach tai khoan ===
```

```
[1] Tai khoan A
```

```
[2] Tai khoan B
```

```
...
```

```
Moi ban chon tai khoan su dung: _
```

- Sau khi người dùng chọn tài khoản cần sử dụng, màn hình hiển thị menu con tương ứng như sau:

```
===== MENU CHUONG TRINH =====
```

```
+ 1. Kiem tra tai khoan.                +
```

```
+ 2. Nap tien vao tai khoan.            +
```

```
+ 3. Rut tien khoi tai khoan.           +
```

```
+ 4. Chuyen tien den tai khoan.         +
```

```
+ 0. Quay ve menu chinh.               +
```

```
=====
```

```
Moi ban chon: _
```

- Thực hiện các chức năng tương ứng với menu con.

-----&📖&-----

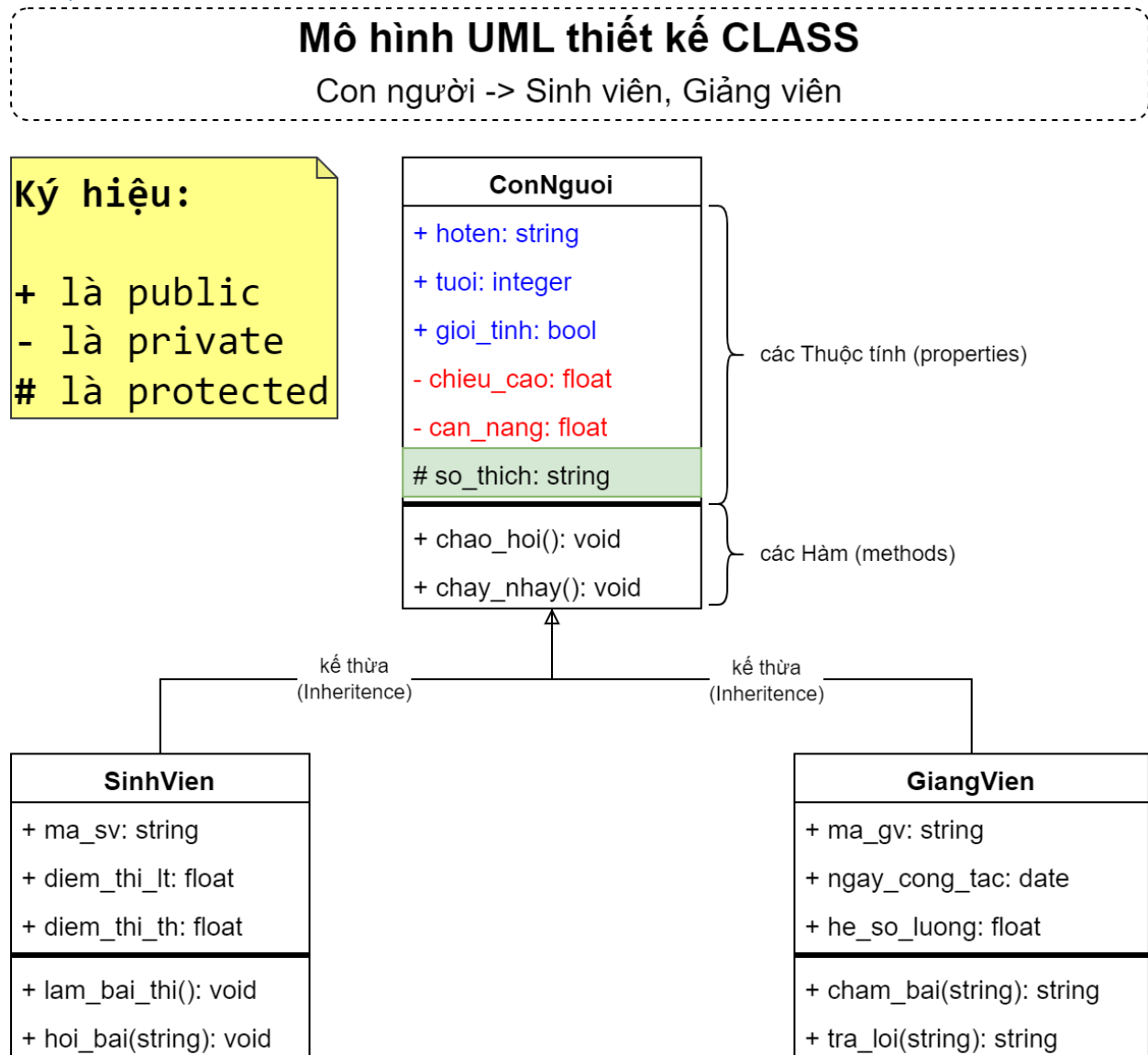


## Chương 2: KẾ THỪA (INHERITANCE), TRỪU TƯỢNG (ABSTRACT), HỢP ĐỒNG (INTERFACE)

### Bài 1: Thiết kế class ConNguoi, SinhVien, GiangVien (100đ)

#### Mục 1: Thông tin

##### a) Sơ đồ UML Class



Hình 2.1: UML class-diagram ConNguoi, SinhVien và GiangVien

#### Mục 2: Yêu cầu

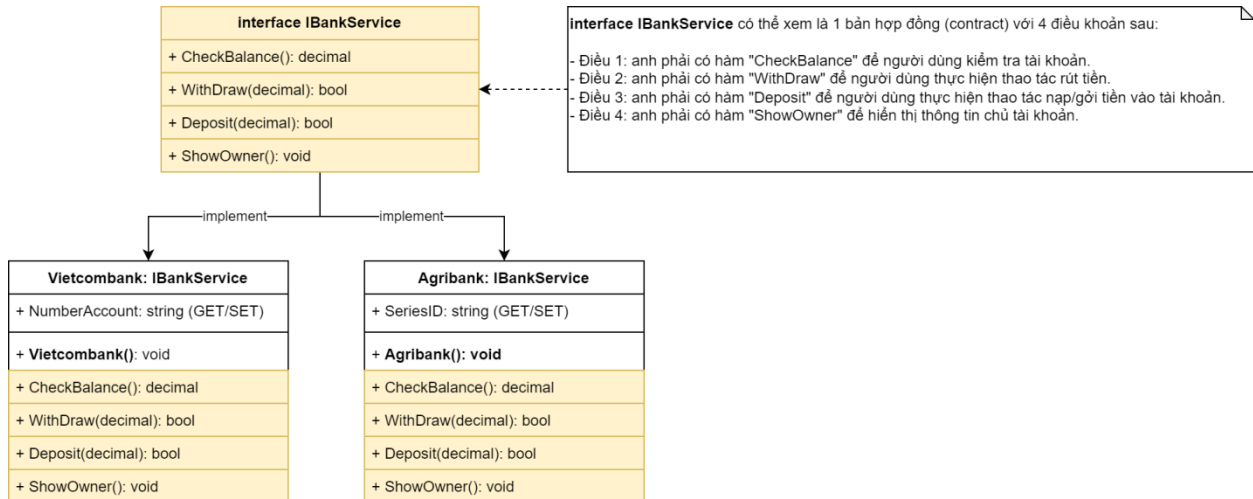
- Xây dựng class theo sơ đồ UML đã thiết kế: (60đ)
- Thực hiện khởi tạo đối tượng (object) từ class vừa thiết kế và thực hiện gọi sử dụng các hàm từ đối tượng đó. (40đ)



## Bài 2: Thiết kế interface IBankService (100đ)

### Mục 1: Thông tin

#### a) Sơ đồ UML Class



Hình 2.2: UML class-diagram IBankService và Vietcombank, Agribank

### Mục 2: Yêu cầu

- Xây dựng class theo sơ đồ UML đã thiết kế: (60đ)
- Thực hiện khởi tạo đối tượng (object) từ class vừa thiết kế và thực hiện gọi sử dụng các hàm từ đối tượng đó. (40đ)



## Bài 3: Thiết kế hệ thống ghi nhật ký chương trình Logger (100đ)

### Mục 1: Thông tin

Người dùng mong muốn ghi log nhật ký theo nhiều phương thức khác nhau:

- Ghi log vào file tập tin.
- Ghi log vào database.
- Ghi log bằng cách thuê đơn vị bên ngoài giám sát.

### Mục 2: Yêu cầu

- Vẽ sơ class theo sơ đồ UML đã thiết kế: (60đ)
- Thực hiện xây dựng các class, khởi tạo đối tượng (object) từ class vừa thiết kế. (chỉ cần xây dựng thực hiện các hàm mô phỏng, không cần cần thực hiện thực tế). (40đ)



## Bài 4: Thiết kế hệ thống quản lý tài khoản ngân hàng của các khách hàng (100đ)

### Mục 1: Thông tin

Ngân hàng thì thường xuyên tổ chức các sự kiện (event) để thu hút khách hàng tham dự.

Ngân hàng có phát hành nhiều loại tài khoản, mỗi khách hàng khi đăng ký thì được phép tạo các loại tài khoản sau:

- Tài khoản thường (checking account).
- Tài khoản tiết kiệm (savings account).

### Mục 2: Yêu cầu

- a. Vẽ sơ class theo sơ đồ UML đã thiết kế: (60đ)
- b. Thực hiện xây dựng các class, khởi tạo đối tượng (object) từ class vừa thiết kế. (chỉ cần xây dựng thực hiện các hàm mô phỏng, không cần cần thực hiện thực tế). (40đ)





## TÀI LIỆU THAM KHẢO

- [1] D. N. P. Cường, "Lập trình Java căn bản," NenTang, 11 06 2023.  
[Online]. Available: <https://nentang.vn/app/edu/khoa-hoc/java/lap-trinh-java-can-ban>. [Accessed 10 09 2024].

**XIN CẢM ƠN !**



**NỀN TẢNG**  
**Hành trang tới Tương Lai**

**Nền tảng Kiến thức - Hành trang tới Tương lai**

**Cung cấp kiến thức nền tảng về Lập trình, Cấu trúc giải thuật, Thiết kế Web, Cơ sở dữ liệu, Thiết kế di động...**

**Giúp các bạn có niềm tin, hành trang kiến thức vững vàng trên con đường trở thành Nhà phát triển Phần mềm trong tương lai!**

**Thông tin liên hệ**

<b>Website</b>	<a href="https://nentang.vn">https://nentang.vn</a>
<b>SĐT + Zalo</b>	0915-659-223
<b>Email</b>	<a href="mailto:phucuong@ctu.edu.vn">phucuong@ctu.edu.vn</a> <a href="mailto:nentangtoituonglai@gmail.com">nentangtoituonglai@gmail.com</a>
<b>Kênh Youtube</b>	<a href="https://www.youtube.com/@nentangtoituonglai">https://www.youtube.com/@nentangtoituonglai</a>