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

MỤC LỤC

[MỤC LỤC I](#_Toc178041834)

[DANH MỤC HÌNH II](#_Toc178041835)

[DANH MỤC BẢNG III](#_Toc178041836)

[DANH MỤC TỪ VIẾT TẮT IV](#_Toc178041837)

[Chương 1: CLASS, OBJECTS, PACKAGES 1](#_Toc178041838)

[Bài 1: Thiết kế class Circle (100đ) 1](#_Toc178041839)

[Mục 1: Thông tin 1](#_Toc178041840)

[Mục 2: Yêu cầu 1](#_Toc178041841)

[Bài 2: Thiết kế class Rectangle (100đ) 3](#_Toc178041842)

[Mục 1: Thông tin 3](#_Toc178041843)

[Mục 2: Yêu cầu 3](#_Toc178041844)

[Bài 3: Thiết kế class Employee (100đ) 5](#_Toc178041845)

[Mục 1: Thông tin 5](#_Toc178041846)

[Mục 2: Yêu cầu 5](#_Toc178041847)

[Bài 4: Thiết kế class BankAccount (100đ) 7](#_Toc178041848)

[Mục 1: Thông tin 7](#_Toc178041849)

[Mục 2: Yêu cầu 7](#_Toc178041850)

[Chương 2: KẾ THỪA (INHERITANCE), TRỪU TƯỢNG (ABSTRACT), HỢP ĐỒNG (INTERFACE) 10](#_Toc178041851)

[Bài 1: Xây dựng hàm tính trung bình tổng của các số tự nhiên chia hết cho 2 (100đ) 10](#_Toc178041852)

[Chương 3: FILES INPUT OUTPUT JSON 11](#_Toc178041853)

[Bài 1: Xây dựng hàm tính trung bình tổng của các số tự nhiên chia hết cho 2 (100đ) 11](#_Toc178041854)

[XIN CÁM ƠN ! B](#_Toc178041855)

DANH MỤC HÌNH

[Hình 1.1: UML cLass-diagram Circle 1](#_Toc178041856)

[Hình 1.2: UML class-diagram Rectangle 3](#_Toc178041857)

[Hình 1.3: UML class-diagram Employee 5](#_Toc178041858)

[Hình 1.4: UML class-diagram BankAccount 7](#_Toc178041859)

DANH MỤC BẢNG

[Bảng 1.1: Kịch bản test case BankAccount 8](#_Toc178041860)

DANH MỤC TỪ VIẾT TẮT

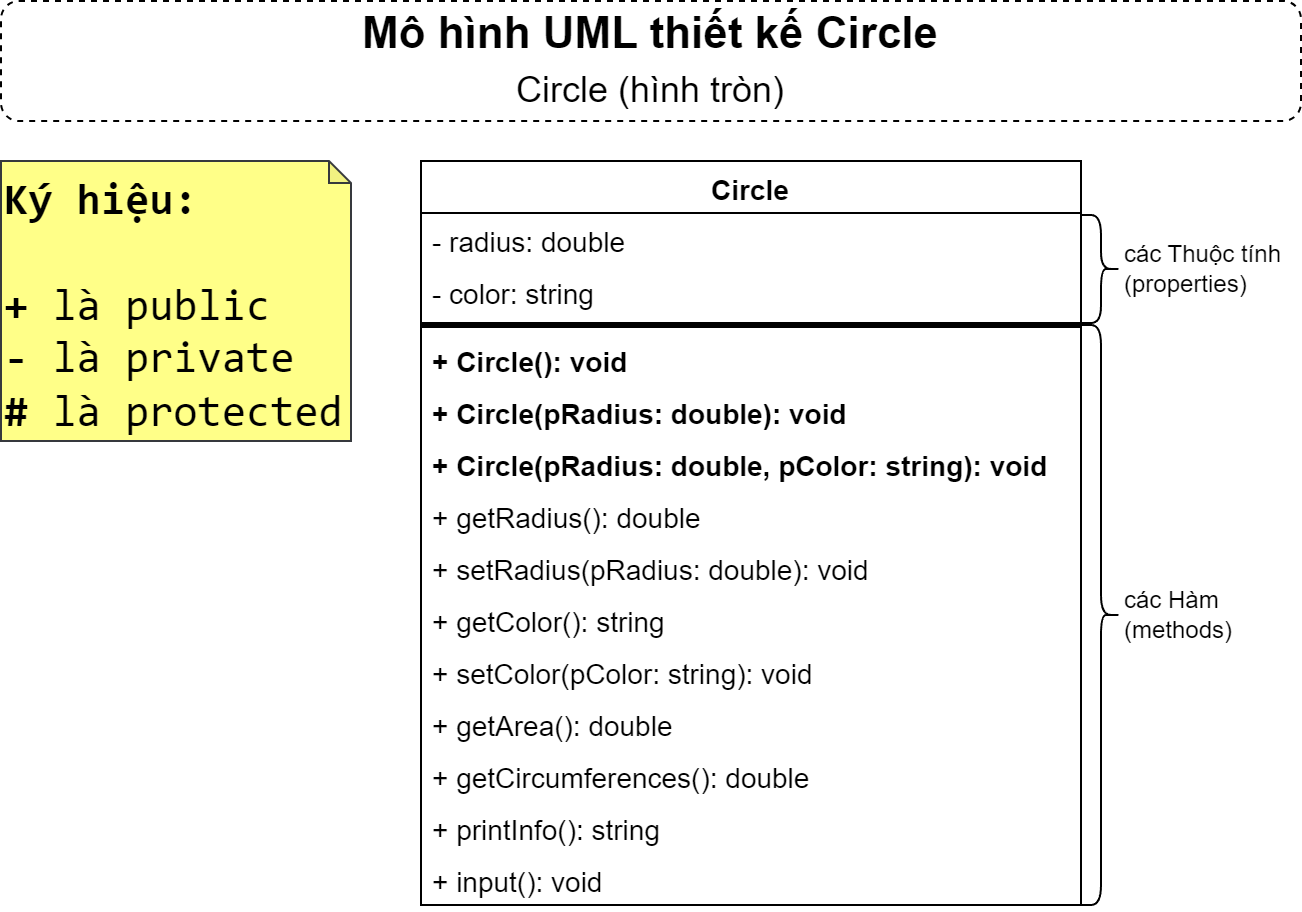
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Từ viết tắt | Tiếng Việt | Tiếng Anh |
|  | CSDL | Cở sở dữ liệu | Database |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# CLASS, OBJECTS, PACKAGES

## Thiết kế class Circle (100đ)

### Thông tin

#### Sơ đồ UML Class



Hình .: UML cLass-diagram Circle

#### Công thức

* Công thức tính diện tích hình tròn:
* Công thức tính chu vi hình tròn:

### Yêu cầu

1. 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: raduis=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.

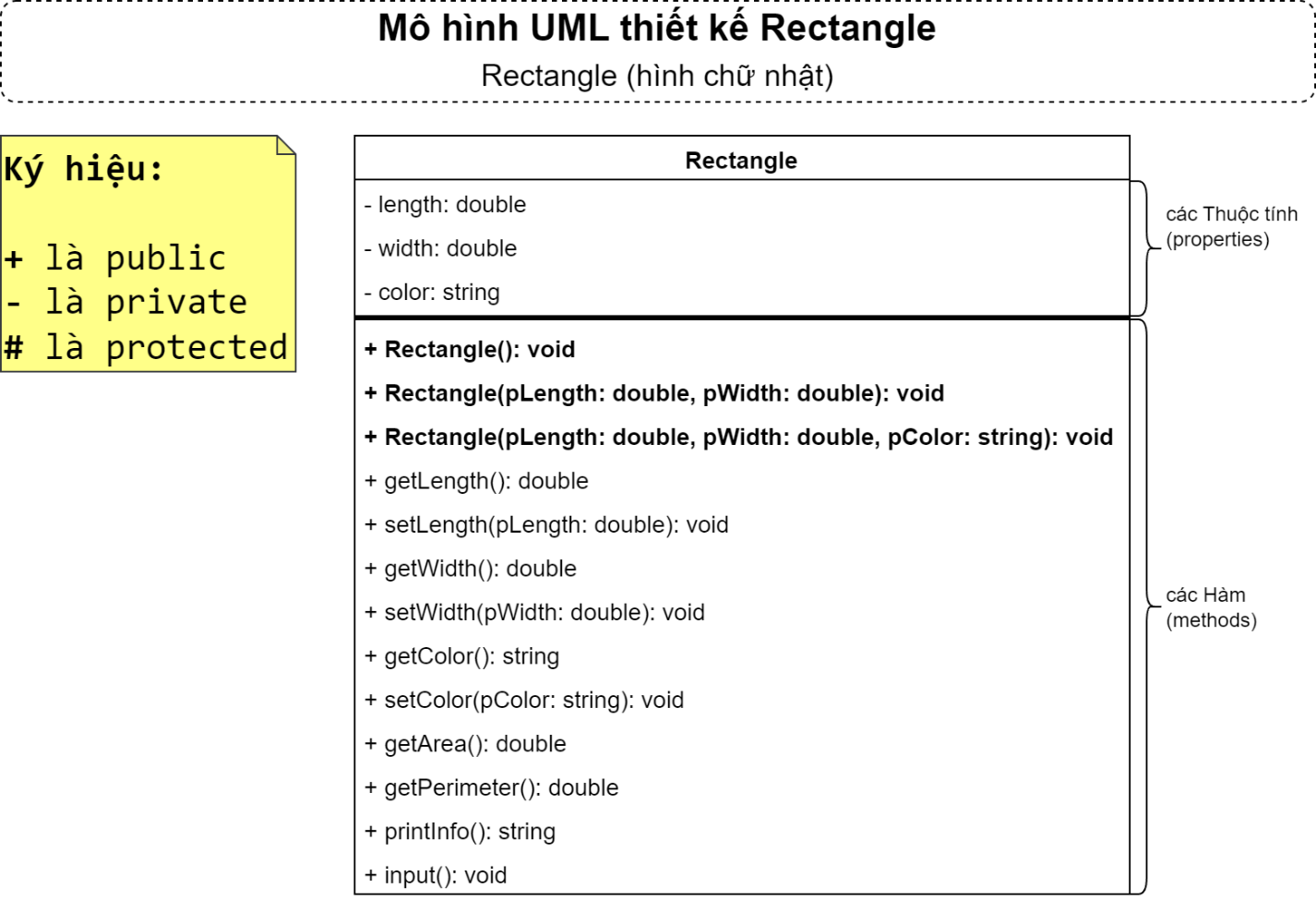
1. Thực hiện khởi tạo đối tượng (object) từ class vừa thiết kế và thực hiện gọi sử dụng các hàm từ đối tượng đó. (40đ)

------🙠🕮🙢------

## Thiết kế class Rectangle (100đ)

### Thông tin

#### Sơ đồ UML Class



Hình .: UML class-diagram Rectangle

#### Công thức

* Công thức tính diện tích hình chữ nhật:
* Công thức tính chu vi hình chữ nhật:

### Yêu cầu

1. 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.

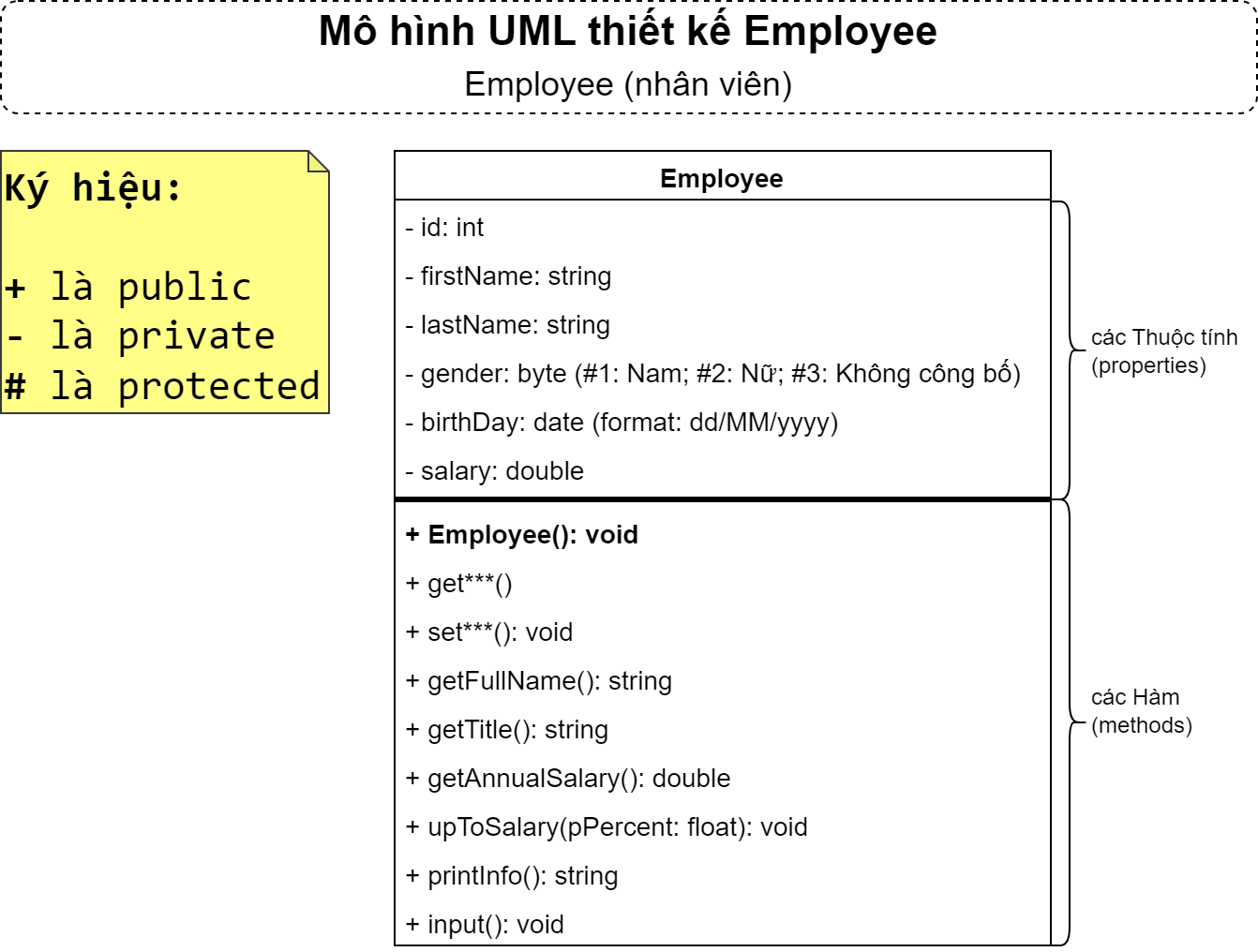
1. Thực hiện khởi tạo đối tượng (object) từ class vừa thiết kế và thực hiện gọi sử dụng các hàm từ đối tượng đó. (40đ)

------🙠🕮🙢------

## Thiết kế class Employee (100đ)

### Thông tin

#### Sơ đồ UML Class



Hình .: UML class-diagram Employee

### Yêu cầu

1. 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 <= 18: gọi là Miss.
    - Nếu tuổi > 18: gọi là Ms.
  + Nếu giới tính là Nam:
    - Nếu tuổi <= 35: gọi là Mr.
    - Nếu tuổi > 35: gọi là Sir.
* Hàm getAnnualSalary() dùng để tính tổng lương hàng năm: salary \* 12 tháng.
* Hàm upToSalary(pPercent) dùng để tăng lương theo phần trăm: salary \* pPercent.
* Hàm printInfo() dùng để in thông tin ra màn hình.
* Hàm input() dùng để thu thập thông tin từ màn hình nhập liệu.

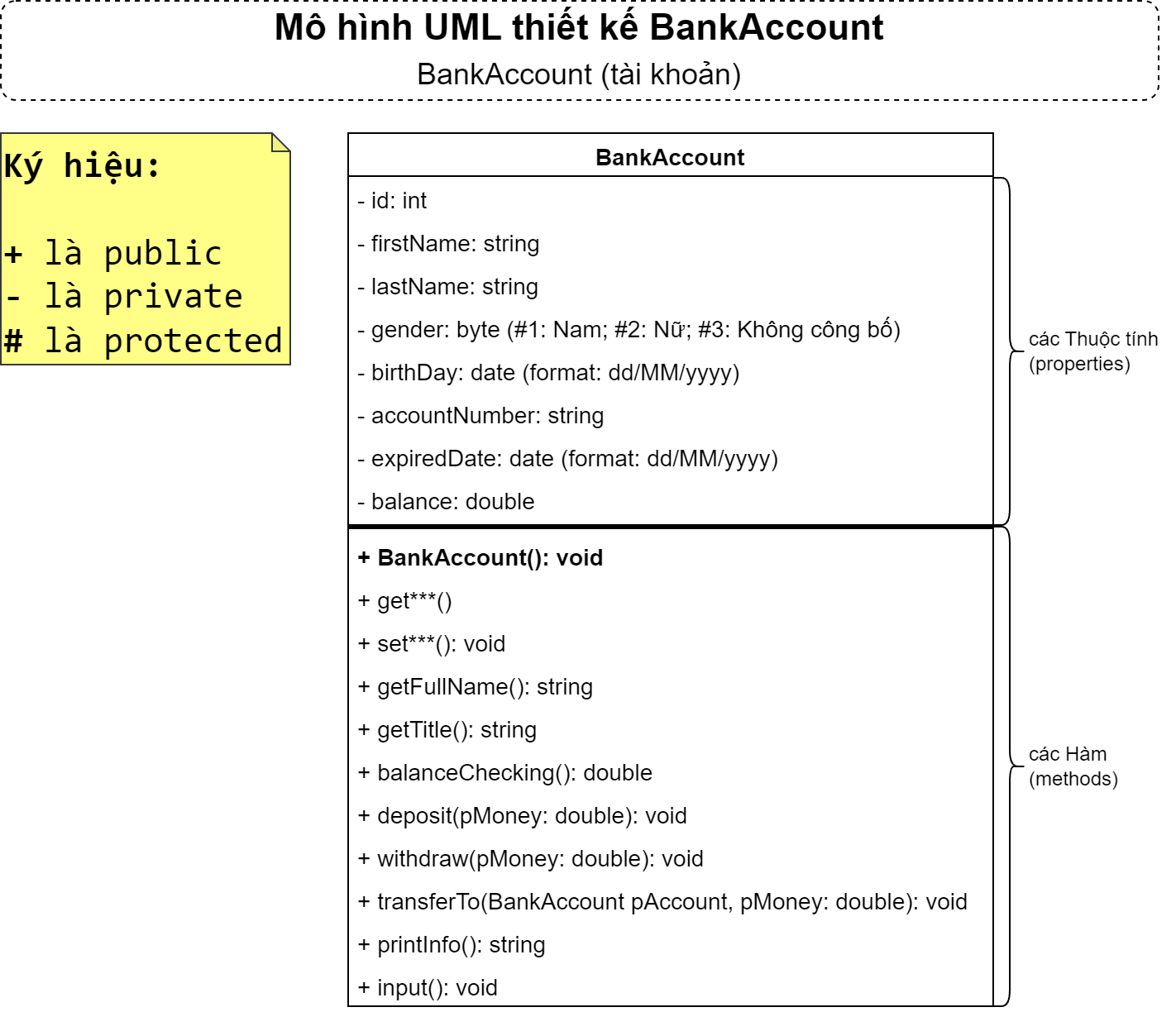
1. Thực hiện khởi tạo đối tượng (object) từ class vừa thiết kế và thực hiện gọi sử dụng các hàm từ đối tượng đó. (40đ)

------🙠🕮🙢------

## Thiết kế class BankAccount (100đ)

### Thông tin

#### Sơ đồ UML Class



Hình 1.4: UML class-diagram BankAccount

### Yêu cầu

1. 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 <= 18: gọi là Miss.
    - Nếu tuổi > 18: gọi là Ms.
  + Nếu giới tính là Nam:
    - Nếu tuổi <= 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.

1. 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)
2. 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 .: Kịch bản test case BankAccount

------🙠🕮🙢------

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

## Xây dựng hàm tính trung bình tổng của các số tự nhiên chia hết cho 2 (100đ)

# FILES INPUT OUTPUT JSON

## Xây dựng hàm tính trung bình tổng của các số tự nhiên chia hết cho 2 (100đ)

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 CÁM ƠN !

------🙠🕮🙢------



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

|  |  |
| --- | --- |
| Website |  |
| **SĐT + Zalo** |  |
| **Email** |  |
| **Kênh Youtube** |  |