

PASSERELLES NUMERIQUES VIETNAM



BÁO CÁO DỰ ÁN QUẢN LÝ LƯƠNG NHÂN VIÊN CỦA MỘT CÔNG TY



Giáo viên: Lê Thị Hồng Thủy

Sinh viên: Hà Mộng Khang & Đặng Văn Đức

Lớp: PNV22A

Đà Nẵng, ngày 29 tháng 10 năm 2020

MỤC LỤC

Lời mở đầu	5
Phần A: LÝ THUYẾT	7
I: Lập trình hướng đối tượng.....	7
1. Định nghĩa	7
2. Đối tượng(object) là gì?.....	7
3. Thuộc tính(attribute) là gì ?.....	8
4. Phương thức(method) là gì ?.....	8
II: Tính trừu tượng hóa.....	9
1. Định nghĩa	9
2. Mối quan hệ giữa các đối tượng.....	10
3. Cách xác định và tối ưu hóa đối tượng trong bài toán.....	13
1. Định nghĩa	14
2. Cơ sở hình thành mối quan hệ kế thừa.....	14
3. Tác dụng của tính kế thừa.....	14
4. Các ví dụ phân tích.....	15
4.1 Bài toán: Quản lý thông tin nhân viên và học sinh của trường tiểu học.....	15
4.2. Các vấn đề phát sinh.....	18
4.3 Ghi đè phương thức(Override).....	21
5.2. Phương thức trừu tượng.....	23
5.3. Ví dụ phân tích	24
IV: TÍNH ĐÓNG GÓI VÀ CHE DẤU DỮ LIỆU.....	29
1. Tính đóng gói.....	29
2. Cách xây dựng các gói(package) chương trình.....	29
3. Tính che dấu dữ liệu	31
4. Tác dụng của tính che dấu dữ liệu	32
5. Cách định nghĩa quyền riêng tư cho đối tượng	33
5.1. Cú pháp khai báo quyền riêng tư.....	33
5.2. Bảng ý nghĩa quyền riêng tư và phạm vi truy cập.....	34
5.3. Quy tắc khai báo quyền riêng tư cho các thuộc tính, phương thức của đối tượng.....	35
6. Vấn đề phát sinh.....	37
V: TÍNH ĐA HÌNH.....	39
1. Định nghĩa	39

2. Ý nghĩa của đa hình	39
2.1. Quản lý danh sách đối tượng có mối quan hệ kế thừa.....	40
a. Định nghĩa	40
b. Từ khóa “instanceOf”	41
c. Vấn đề phát sinh	41
2.2. Quản lý bộ hành vi của các đối tượng không cùng bản chất	42
a. Định nghĩa	42
b. Từ khóa “interface”	42
c. Từ khóa “implement”	43
PHẦN B: DỰ ÁN.....	44
I. GIỚI THIỆU DỰ ÁN:.....	44
1. Mục đích.....	44
2. Thực tế hiện nay	44
3. Hệ thống được đề xuất.....	45
II. MÔ TẢ TỔNG THỂ.....	45
1. Người sử dụng hệ thống.....	45
2. Hệ thống chức năng.....	45
III. PHÂN TÍCH YÊU CẦU	47
1. Thông tin và yêu cầu của khách hàng:.....	47
2. Thông tin qua khảo sát thực tế:	47
3. Mô tả hệ thống phần mềm tính lương.....	48
Quy trình tính lương của công ty:.....	48
a. Cách tính lương:	48
4. Đặc tả bài toán tính lương nhân viên	49
IV. GIẢI PHÁP	50
1. Lựa chọn ngôn ngữ lập trình: <i>Ngôn ngữ lập trình Java</i>	50
2. Bảng tiến độ và phân công công việc:.....	50
3. Mô hình xây dựng:.....	52
a. Sơ đồ use case:	52
Mô tả sơ đồ use case:	53
b. Sơ đồ class diagram:	58
4: Giao diện ứng dụng:.....	59
a. Tài khoản đăng nhập:	59

b. Giao diện Menu chính:.....	60
c. Giao diện nhập thông tin quản lý:.....	61
d. Giao diện menu xem thông tin:	62
e. Thêm nhân viên:	63
f. Giao diện chức năng menu quản lý lương:	64
g. Giao diện menu tìm nhân viên:	65
PHẦN C: KIẾN THỨC VÀ KINH NGHIỆM ĐẠT ĐƯỢC.....	65
PHẦN D: KẾT LUẬN	68
Danh sách link:	68

Lời mở đầu

Chào tất cả các bạn yêu thích lập trình!

Tôi là Đặng Văn Đức, tôi đến từ Quảng Bình, hiện tại tôi đang là sinh viên IT năm 2 tại tổ chức PASSERELLES NUMERIQUES. Mục đích tôi có mặt tại đây là để giúp cho các bạn có thể hiểu thêm một số kiến thức quan trọng trong lập trình hướng đối tượng trong IT.

Java là một công nghệ mở được xây dựng dựa trên tư duy hướng đối tượng, được phát triển bởi tập đoàn Oracle và đang ngày càng phát triển hiện nay. Việc ứng dụng của công nghệ này trong các lĩnh vực Công Nghệ Thông Tin như: Cloud, Web, Mobile, BigData, AI, IoT là rất đa dạng, do đó nó mang đến rất nhiều cơ hội việc làm, đặc biệt là ở các thành phố trọng điểm tại Việt Nam hiện nay như Hà Nội, Đà Nẵng, Thành phố Hồ Chí Minh,...

Lập trình hướng đối tượng Java là một sự định hướng đúng đắn cho những bạn trẻ đang mới bắt đầu tìm tòi để trở thành một lập trình viên.

Những gì mà tôi trình bày dưới đây phù hợp với đa phần những bạn còn chưa vững chắc một số kiến thức căn bản về tư duy lập trình hướng đối tượng và một số kiến thức nền tảng trong Java. Hầu hết các sinh viên công nghệ thông tin, đặc biệt là các sinh viên lập trình đều rất yếu về tư duy giải quyết một bài toán.

Các bạn thân mến, để giải quyết một vấn đề, trước tiên chúng ta cần phải có phương pháp, tiếp theo là sử dụng phương pháp đó để mổ xẻ vấn đề, phân tích chúng cho tường minh, rõ ràng, từ đó mới có thể xác định được chính xác các bước cần thực hiện.

Và một phương pháp tối ưu, tiên tiến, phù hợp để giải quyết tất cả các vấn đề lớn nhỏ, các bài toán từ đơn giản đến phức tạp như các phần mềm quản lý, các loại game 2D, 3D,... đó là **Lập trình hướng đối tượng**.

Bài viết này được biên soạn dựa trên kinh nghiệm trau dồi kiến thức từ bạn bè và thầy cô mà bản thân tôi đúc rút được trong khoảng thời gian tôi học tại tổ chức PASSERELLES NUMERIQUES. Bài viết sẽ cập nhật một số kiến thức trong Lập trình hướng đối tượng và được trình bày một cách khá chi tiết, và dễ hiểu nhất. Bên cạnh đó, sẽ có một dự án kèm theo để giúp cho các bạn hình dung tốt hơn có tên là: **Quản lý lương nhân viên** dưới sự trình bày tôi và một người bạn thân của tôi là Hà Mộng Khang.

Mọi chi tiết, đóng góp xin vui lòng liên hệ chúng tôi qua những địa chỉ thông tin sau:

Tác giả: **Hà Mộng Khang** và **Đặng Văn Đức**

Email: duc.dang22@student.passerellesnumeriques.org

Số điện thoại: 0902312207

Chúc các bạn học tốt!

Xin chân thành cảm ơn!

Phần A: LÝ THUYẾT

I: Lập trình hướng đối tượng

1. Định nghĩa

- Là phương pháp phân tích bài toán tiên tiến, được phát triển sau phương pháp hướng chức năng.
- Phương pháp hướng đối tượng, mỗi bài toán sẽ phân tích thành các đối tượng, mỗi đối tượng bao gồm đặc điểm và hành vi.
- Để giải quyết bài toán, chúng ta đi xây dựng mối quan hệ giữa các đối tượng và cho từng đối tượng thực thi hành vi của chính nó.
- Đây là phương pháp tư duy gần với thực tế cuộc sống của con người, xây dựng bài toán dựa trên mối quan hệ giữa các đối tượng tham gia vào bài toán.
- Phù hợp với các bài toán phức tạp, khi không thể phân tích được hết các công việc cần làm.
- Dữ liệu của bài toán chính là các đặc điểm của từng đối tượng, nên sẽ hình thành được cơ chế bảo mật cho dữ liệu.
- Tái sử dụng được các tài nguyên cho các bài toán khác nhau.

2. Đối tượng(object) là gì?

- Đối tượng là một thực thể sống tham gia vào bài toán, có đặc điểm và hành vi.
- Là một danh từ, và không thể xác định được miền giá trị.

Ví dụ: học sinh, con mèo, xe máy,...

Các danh từ vừa liệt kê là các đối tượng, vì chúng đều có cấu tạo phức tạp, không thuộc bất cứ miền giá trị có sẵn nào.

3. Thuộc tính(attribute) là gì ?

- Là các đặc điểm của đối tượng.
- Là một danh từ có một giá trị cụ thể và có miền giá trị đã được xác định.

Ví dụ: Nguyễn Văn Anh, 123, 123.5, ...

4. Phương thức(method) là gì ?

- Là một hành vi của đối tượng.
- Là một động từ mô tả hành động của đối tượng trong bài toán.
- Đối với mỗi đối tượng, ngoài các phương thức được liệt kê trong bài toán, luôn có thêm 2 phương thức nữa là nhập thông tin và in thông tin.

+ Nhập thông tin: Là hành vi nhập giá trị đầu vào cho các thuộc tính của đối tượng.

+ In thông tin: Là hành vi in ra các thông tin của đối tượng cho người dùng biết.

Ví dụ về một số phương thức:

+ Giới thiệu về bản thân: là một hành vi của một ai đó

+ Tính điểm trung bình: là một hành vi của đối tượng học sinh để tính giá trị điểm trung bình của các môn học,...

II: Tính trừu tượng hóa

1. Định nghĩa

- Trừu tượng hóa là việc loại bỏ đi hoặc không chú ý đến một số khía cạnh, đặc tính của đối tượng mà tập trung vào các đặc điểm cốt lõi của đối tượng được đề cập trong bài toán, nhằm làm nổi bật vai trò của đối tượng và tránh dư thừa dữ liệu.

Chú ý: Nếu các đối tượng được đề cập đến trong bài toán không có các thông tin đặc điểm cụ thể thì chúng ta bắt buộc phải áp dụng tính trừu tượng hóa để đưa ra một số các đặc điểm quan trọng và cần thiết của đối tượng phù hợp với vai trò của đối tượng được nhắc đến trong bài toán

Bài toán: Quản lý điểm Toán, Văn, Anh của học sinh tiểu học.

Người quản lý có thể thêm học sinh vào lớp để quản lý, in thông tin các học sinh trong lớp, xóa học sinh, tìm kiếm học sinh theo tên, sắp xếp học sinh tang dần theo điểm trung bình. Thông tin bao gồm: Các thông tin cơ bản, điểm Toán, điểm Văn, điểm Anh.

❖ Phân tích

Cùng áp dụng tư duy hướng đối tượng vào bài toán trên thì có các đối tượng gồm **Học sinh, Quản lý**

Ở đây khi phân tích đối tượng **Học sinh**, sẽ gồm các thuộc tính: Các thông tin cơ bản, điểm Toán, điểm Anh, điểm Văn.

Ta có thể thấy bài toán đưa ra trong trường hợp này là không rõ ràng về các thông tin cơ bản của đối tượng **Học sinh** . Do đó, chúng ta sẽ phải phân tích và đưa ra một số đặc điểm cốt lõi để đại diện cho thông tin cơ bản của **Học sinh**.

Nếu không áp dụng tính trừu tượng, chúng ta sẽ đưa ra vô số các đặc điểm về **Học sinh** như sau: Họ tên, giới tính, sở thích, tên lớp, mã học sinh, địa chỉ, số điện thoại, họ tên cha, họ tên mẹ, chiều cao, cân nặng,...

Tuy nhiên, phần lớn các đặc điểm nêu trên là không cần thiết.

Khi trừu tượng hóa, chúng ta tập trung vào các đặc điểm cốt lõi của đối tượng **Học sinh** đang được đề cập đến trong bài toán chỉ gồm các thông tin sau: Họ tên, giới tính, mã học sinh, tên lớp.

2. Mối quan hệ giữa các đối tượng

❖ Định nghĩa

- Mối quan hệ quy định sự ràng buộc và khả năng chia sẻ dữ liệu giữa các đối tượng.
- Việc xác định mối quan hệ giữa các đối tượng có mặt trong bài toán sẽ giúp:
 - Phân tích các đối tượng tối ưu hơn, độc lập nhau.
 - Hình thành kiến trúc chương trình và luồng hoạt động giữa các đối tượng một cách tối ưu, tường minh, rõ ràng.

❖ Các mối quan hệ

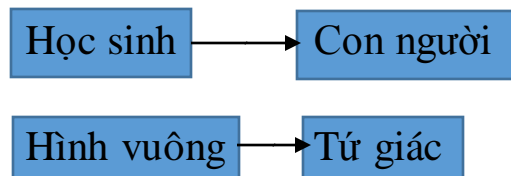
➤ Quan hệ kế thừa

- Nếu đối tượng A thừa hưởng lại các đặc điểm và hành vi của đối tượng C mà không cần phải khai báo thì ta nói: Giữa A và C có mối quan hệ kế thừa. A kế thừa C.

Kí hiệu: 

- Nếu đối tượng A và đối tượng B có cùng bản chất, cùng chung một số đặc điểm và hành vi, thì hình thành đối tượng C gồm các đặc điểm chung của hai đối tượng A và B. Khi đó đối tượng A và B sẽ kế thừa đối tượng C.
- Sử dụng mối quan hệ kế thừa trong phân tích sẽ giúp tạo ra danh sách các đối tượng tối ưu và kiến trúc code trở nên tường minh, mạch lạc.

Ví dụ mối quan hệ kế thừa:



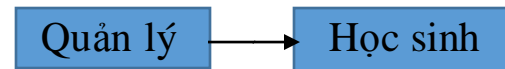
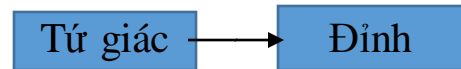
➤ Quan hệ phụ thuộc

- Nếu đối tượng B trở thành thuộc tính của đối tượng A thì nói A phụ thuộc vào B.

Kí hiệu: 

- Nếu thuộc tính A₁ của đối tượng A có miền giá trị không có sẵn trong tự nhiên thì miền giá trị của thuộc tính A₁ phải là một đối tượng cần được định nghĩa. Khi đó cần:
 - Xây dựng đối tượng B để định nghĩa miền giá trị cho thuộc tính A₁.
 - Khai báo thuộc tính A₁ thuộc miền giá trị B

Ví dụ mối quan hệ phụ thuộc:



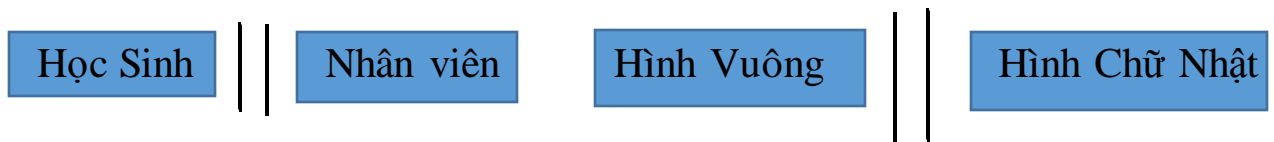
Giải thích:

Trong Tứ Giác có 4 đỉnh A,B,C,D. Tuy nhiên các thuộc tính A,B,C,D này lại có miền giá trị là đỉnh. Do đó cần:

- Xây dựng đối tượng **Đỉnh** gồm có thuộc tính x,y và một số phương thức.
- Khai báo thuộc tính A,B,C,D thuộc miền giá trị **Đỉnh**.

➤ Quan hệ song song

- Nếu đối tượng A và đối tượng B độc lập nhau, không có sự ràng buộc nào thì ta nói A và B song song nhau.



- Trong mỗi quan hệ song song, tuy A và B không liên quan đến nhau nhưng có thể trao đổi dữ liệu cho nhau.

Ví dụ: Học sinh A và học sinh B trong 1 lớp học là hai đối tượng song song nhau, tuy nhiên lại có thể trao đổi dữ liệu cho nhau.

3. Cách xác định và tối ưu hóa đối tượng trong bài toán

Bước 1: Xác định các đối tượng có mặt trong bài toán.

Bước 2: Áp dụng trừu tượng hóa để tối ưu các thuộc tính của đối tượng.

Bước 3: Loại bỏ các thuộc tính và phương thức thừa như sau:

- Nếu một thuộc tính mà không được sử dụng trong bất kì một phương thức nào của đối tượng thì thuộc tính đó là thuộc tính thừa, cần loại bỏ.
- Nếu một phương thức mà không sử dụng bất kì một thuộc tính nào để hoạt động thì phương thức đó bị thừa, cần loại bỏ.

Bước 4: Xây dựng mối quan hệ giữa các đối tượng:

- Tìm kiếm mối quan hệ kế thừa:
 - Nếu đối tượng A và đối tượng B có cùng bản chất, cùng chung1 số đặc điểm và hành vi, thì hình thành lên đối tượng cha C gồm các đặc điểm chung của hai đối tượng A và B.
 - Xóa tất cả các đặc điểm chung, hành vi chung của A và B.
- Tìm mối quan hệ phụ thuộc.

III: TÍNH KẾ THỪA

1. Định nghĩa

- Kế thừa là sự thừa hưởng lại các thuộc tính , phương thức của các đối tượng con từ một đối tượng cha mà không cần phải khai báo thuộc tính, phương thức đó.
- Trong java không có đa kế thừa. Mỗi một đối tượng chỉ được phép phép kế thừa trực tiếp từ một đối tượng khác.
- Sử dụng từ khóa **extends** để hiển thị mối quan hệ kế thừa.
Ví dụ: HocSinh **extends** ConNguoi

2. Cơ sở hình thành mối quan hệ kế thừa

- Nếu đối tượng A và đối tượng B có cùng bản chất, cùng chung 1 số đặc điểm và hành vi, thì hình thành lên đối tượng C gồm các đặc điểm chung của hai đối tượng A và B. Khi đó đối tượng A và B sẽ kế thừa đối tượng C.

3. Tác dụng của tính kế thừa

- Sử dụng mối quan hệ kế thừa trong phân tích sẽ giúp tạo ra danh sách các đối tượng tối ưu và kiến trúc code trở nên tường minh, mạch lạc, giúp cho việc bảo trì và nâng cấp chương trình được dễ dàng.

4. Các ví dụ phân tích

4.1 Bài toán: Quản lý thông tin nhân viên và học sinh của trường tiểu học

- Các đối tượng có mặt trong bài toán: **Học Sinh, Nhân Viên, Quản Lý.**

❖ Học Sinh:

- Thuộc tính: Mã HS, Họ Tên, Lớp, Tuổi, Giới Tính, Địa Chỉ, Điểm Toán, Điểm Anh, Điểm Văn.
- Phương thức:
 - Nhập thông tin
 - In thông tin
 - Học bài
 - Ăn uống
 - Di chuyển
 - Tính điểm trung bình

❖ Nhân Viên:

- Thuộc tính: Mã NV, Họ Tên, Chức Vụ, Địa Chỉ, Tuổi, Giới Tính, Lương.
- Phương thức:
 - Nhập thông tin
 - In thông tin
 - Làm việc
 - Ăn uống
 - Di chuyển

- Viết báo cáo

❖ Quản Lý:

- Thuộc tính: Danh sách<Nhân Viên>, danh sách<Con Người>.
- Phương thức:
 - Thêm nhân viên
 - Thêm học sinh
 - Tìm kiếm nhân viên
 - Tìm kiếm học sinh
 - Xóa nhân viên
 - Xóa học sinh
 - Hiện thị danh sách nhân viên
 - Hiện thị danh sách học sinh.

- Tìm kiếm mối quan hệ kế thừa:

Hai đối tượng Học Sinh và Nhân Viên, có các thuộc tính chung: Họ Tên, Giới Tính, Địa Chỉ, Tuổi

Hai đối tượng Học Sinh và Nhân Viên còn có các phương thức chung như:

- Nhập thông tin
- Ăn uống
- Di chuyển
- In thông tin

- **Từ đó suy ra:**

Ta hình thành đối tượng cha có tên Con Người chứa các đặc điểm chung của Học Sinh và Nhân Viên. Vậy bài toán có 4 đối tượng cần quản lý đó là: Con Người, Học Sinh, Nhân Viên, Quản Lý.

❖ **Con Người:**

- Thuộc tính: Họ Tên, Tuổi, Giới Tính, Địa Chỉ
- Phương thức:
 - Nhập thông tin
 - Ăn uống
 - Di chuyển
 - In thông tin

❖ **Học Sinh:**

- Thuộc tính: Mã HS, Lớp, Điểm Toán, Điểm Văn, Điểm Anh.
- Phương thức:
 - Học bài
 - Tính điểm trung bình

❖ **Nhân Viên:**

- Thuộc tính: Mã NV, Chức Vụ, Lương
- Phương Thức:
 - Làm việc
 - Viết báo cáo

❖ **Quản Lý:**

- Thuộc tính: Danh sách<Nhân Viên>, Danh Sách<Con Người>

- Phương thức:
 - Thêm nhân viên
 - Thêm học sinh
 - Tìm kiếm nhân viên
 - Tìm kiếm học sinh
 - Xóa nhân viên
 - Xóa học sinh
 - Hiện thị ds nhân viên
 - Hiện thị ds học sinh

4.2. Các vấn đề phát sinh

❖ Vấn đề 1: Nạp chồng phương thức trong đối tượng con

```
void nhapTT( String hoTen, String gioiTinh, String diaChi, int tuoi){
    //Nhap thông tin cho các thuộc tính hoTen, gioiTinh, diaChi, tuoi
}
```

- Phương thức này được xây dựng trong đối tượng cha và được các đối tượng con sử dụng, tuy nhiên nó không đầy đủ vì nó chỉ có thể nhập được thông tin cho các thuộc tính: họ tên, giới tính, địa chỉ, tuổi.
- Đối với **Học Sinh** còn có thêm 5 thuộc tính riêng nữa : mã học sinh, lớp, điểm Toán, điểm Anh, điểm Văn, do đó nếu chỉ sử dụng phương thức của đối tượng cha là **Con Người** thì không đủ để thực hiện được hết các công việc cần làm cho đối tượng con là **Học Sinh**.
- Đối tượng **Nhân Viên** cũng tương tự.
- Do đó, cần nạp chồng phương thức **nhapTT** cho các đối tượng con để nhập thêm dữ liệu cho **Học Sinh** và **Nhân Viên**

```

Class HocSinh extends ConNguoi {
    String maHS, lop;
    int dienToan, diemAnh, diemVan;

    void nhapTT( String hoTen, String gioiTinh, String diaChi, int tuoi, String
maHS, String lop, float diemToan, float diemAnh, float diemVan){
//goi den phuong thuc nhapTT cua doi tuong cha
        nhapTT( hoTen, gioiTinh, diaChi, tuoi);
//Nhap thong tin cho cac thuoc tinh ma HS, diem Toan, diem Van, diem Anh
    }
    ....
}

```

```

Class NhanVien extends ConNguoi{
    String maNV, chucVu;
    double luong;

    void nhapTT( String hoTen, String gioiTinh, String diaChi, int tuoi, String
maNV, String chucVu, double luong){
//goi den phuong thuc nhapTT cua doi tuong cha
        nhapTT( hoTen, gioiTinh, diaChi, tuoi);
//nhap thong tin chon cac thuoc tinh ma NV, chuc vu, luong
    }
    ....
}

```

```
void inTT(){  
    //in thông tin con người gồm họTen, giớiTinh, địaChi, tuổi  
}
```

❖ Vấn đề 2: Ghi đè phương thức trong đối tượng con

- Phương thức này được xây dựng trong đối tượng cha và được các đối tượng con sử dụng, tuy nhiên nó không đầy đủ vì nó chỉ có thể in được thông tin các thuộc tính: họ tên, giới tính, địa chỉ, tuổi.
- Đối với **Học Sinh** còn có thêm 5 thuộc tính riêng nữa : mã học sinh, lớp, điểm Toán, điểm Anh, điểm Văn, do đó nếu chỉ sử dụng phương thức của đối tượng cha là **Con Người** thì không đủ để thực hiện được hết các công việc cần làm cho đối tượng con là **Học Sinh**.
- Đối tượng **Nhân Viên** cũng tương tự.
- Tuy nhiên, liệu có thể nạp chồng phương thức **inTT** cho các đối tượng con để in thêm dữ liệu cho **Học Sinh** và **Nhân Viên** ?
 - **Trả lời:** không thể nạp chồng phương thức **inTT**. Vì phương thức **inTT** không cần tham số đầu vào, do đó không thể xây dựng 2 phương thức **inTT** giống hệt nhau được. (vi phạm nạp chồng).
- Do đó, không thể sử dụng nạp chồng để giải quyết các vấn đề được, chúng ta phải sử dụng **Ghi đè phương thức**.

4.3 Ghi đè phương thức(Override).

- Là việc xây dựng lại nội dung của phương thức đã được định nghĩa hoặc xây dựng trong đối tượng cha cho phù hợp với các công việc cần xử lý của đối tượng con.
- Có hai hình thức ghi đè:
 - **Ghi đè hoàn toàn:** Là việc xóa bỏ và xây dựng lại hoàn toàn nội dung của phương thức cha trong lớp con. Ghi đè hoàn toàn được sử dụng trong hai tình huống sau:
 - Phương thức của đối tượng cha chỉ dừng ở mức định nghĩa, không có nội dung.
 - Phương thức ở đối tượng cha hoạt động kém hiệu quả, không tối ưu. Do đó cần phải xây dựng hoàn toàn lại ở lớp con.
 - **Ghi đè không hoàn toàn:** là việc kế thừa và xây dựng thêm nội dung của phương thức cha trong các lớp con. Ghi đè không hoàn toàn được sử dụng trong bài toán sau:
 - Phương thức của đối tượng cha chưa thỏa mãn các công việc cần xử lý trong các lớp con, do đó các lớp con cần được bổ sung thêm.
 - Sử dụng từ khóa **super.Tên_Phiơng_Thức** khi ghi đè không hoàn toàn để gọi đến phương thức của đối tượng cha để kế thừa lại các công việc đã được đối tượng cha thực hiện.
 - Sử dụng kí hiệu **@Override** để thông báo phương thức nào là phương thức ghi đè trong đối tượng con.

➤ Ta sẽ đi giải quyết vấn đề 2 bằng “ Ghi đè phương thức”.

Ở đây là ghi đè không hoàn toàn vì chúng ta cần tận dụng những gì đối tượng cha đã làm trong phương thức **inTT** để in ra thông tin các thuộc tính chung như họ tên, địa chỉ, giới tính, tuổi.

```
Class HocSinh extends ConNguoi{
    String maHS, lop;
    int diemToan, diemVan, diemAnh;

    @Override
    void inTT(){
        // gọi tên phương thức inTT của đối tượng cha
        Super.inTT();
        //In thêm thông tin cho các thuộc tính ma HS, lop, diemToan, diem
        Van, diem Anh
    }
    ...
}
```

```
Class NhanVien extends ConNguoi{
    String maNV, chucVu;
    double luong;

    @Override
    void inTT(){
        //gọi tên phương thức inTT của đối tượng cha
        Super.inTT();
        //In thêm thông tin cho các thuộc tính ma NV, chuc vu, luong
    }
    ...
}
```

5. Đối tượng trừu tượng, phương thức trừu tượng

5.1. Đối tượng(lớp) trừu tượng

Một lớp trở thành trừu tượng khi:

- Bản thân nó chứa các phương thức trừu tượng nên bắt buộc lớp đó phải trở thành trừu tượng.
- Khi đối tượng này không trực tiếp tham gia vào bài toán, do đó để xây dựng một kiến trúc code tường minh và logic, những đối tượng như vậy chúng ta sẽ định nghĩa nó là lớp trừu tượng.
- Sử dụng từ khóa **abstract** để định nghĩa 1 lớp là lớp trừu tượng.

5.2. Phương thức trừu tượng

- Phương thức trừu tượng là phương thức chỉ có định nghĩa chứ không có nội dung và nằm trong lớp cha.
- Một phương thức trở thành trừu tượng khi:
 - Không thể xây dựng nội dung hoặc rất khó để xây dựng được nội dung cho phương thức đó trong lớp cha. Do đó, trong lớp cha chỉ cần định nghĩa phương thức đó. Các lớp con sẽ chịu trách nhiệm xây dựng nội dung.
 - Phương thức đó bị ghi đè hoàn toàn trong tất cả các lớp con.
- Khi một phương thức trở thành trừu tượng thì:
 - Lớp cha chứa phương thức đó sẽ phải trở thành lớp trừu tượng.
 - Các lớp con bắt buộc phải **ghi đè hoàn toàn** phương thức trừu tượng này của lớp cha.
 - Sử dụng từ khóa **abstract** để định nghĩa một phương thức là trừu tượng.

5.3. Ví dụ phân tích

Bài toán: Quản lý thông tin nhân viên và học sinh của trường tiểu học.

- Bài toán có 4 đối tượng: Con Người, Học Sinh, Nhân Viên, Quản Lý.
 - Đối tượng cha trong trường hợp này là **Con Người** không trực tiếp tham gia vào bài toán. Hơn nữa 2 hành vi **Ăn Uống** và **Di Chuyển** là 2 hành vi sẽ được ghi đè hoàn toàn trong đối tượng **Học Sinh** và **Nhân Viên**, vì **Học Sinh** và **Nhân Viên** đều có cách di chuyển và ăn uống khác nhau.
- Do đó ta có:
- **Con Người** là đối tượng trừu tượng
 - **Ăn Uống** và **Di Chuyển** là 2 hành vi trừu tượng
- ❖ Đối tượng Con Người

```
abstract class ConNguoi{
    String hoTen, gioiTinh, diaChi;
    int tuoi;

    void nhapTT( String hoTen, String gioiTinh, String diaChi, int tuoi){
        //nhap thong tin cho cac thuoc tinh hoTen, gioiTinh, diaChi, tuoi
    }

    //dinh nghĩa anUong la phuong thuc truu tuong
    abstract void anUong();

    // dinh nghĩa diChuyen la phuong thuc truu tuong
    Abstract void diChuyen();

    void inTT(){
        //In thong tin con nguoi gom: hoTen, diaChi, gioiTinh, tuoi
    }
}
```


❖ Đối tượng Học Sinh

```
Class HocSinh extends ConNguoi{
    String maHS, lop;
    int diemToan, diemAnh, diemVan;

    void nhapTT( String hoTen, String gioiTinh, String diaChi, int tuoi,
String maHS, String lop, float diemToan, float diemAnh, float diemVan){
//gọi đến phương thức nhapTT của đối tượng cha
        nhapTT( hoTen, gioiTinh, diaChi, tuoi);
//Nhập thông tin cho các thuộc tính ma HS, diem Toan, diem Van, diem Anh
    }

    void hocBai(){
//Nội dung phương thức học bài
    }

    Double tinhDiemTB(){
//Nội dung phương thức tính điểm trung bình
        Return -1;
    }

    @Override
        Void anUong(){
//Xây dựng hành vi ăn uống của Sinh Viên
//Ghi đề hoàn toàn nên không có super
    }

    @Override
        Void diChuyen(){
//Xây dựng hành vi đi chuyển của Sinh Viên
//Ghi đề hoàn toàn nên không có super
    }
}
```

```

    }

    @Override
    Void inTT(){
        //Goi den phuong thuc inTT cua doi tuong cha
        Super.inTT();
        //In them thong tin cho cac thuoc tinh ma NV, chuc vu, luong
    }
}

```

❖ Đối tượng Nhân Viên

```

Class NhanVien extends ConNguoi{
    String maNV, chucVu;
    Int luong;

    void nhapTT( String hoTen, String gioiTinh, String diaChi, int tuoi,
        String maNV, String chucVu, double luong){
        //goi den phuong thuc nhapTT cua doi tuong cha
        nhapTT( hoTen, gioiTinh, diaChi, tuoi);
        //nhap thong tin chon cac thuoc tinh ma NV, chuc vu, luong
    }

    Void lamViec(){
        //Noi dung phuong thuc lam viec
    }

    Void vietBaoCao(){
        //Noi dung phuong thuc viet bao cao
    }
}

```

```

    }

    @Override
    Void anUong(){
        //xay dung hanh vi an uong cua nhan vien
        //Ghi de hoan toan nen khong co super
    }

    @Override
    Void diChuyen(){
        //Xay dung hanh vi di chuyen cua nhan vien
        //Ghi de hoan toan nen khong co super
    }

    @Override
    Void diChuyen(){
        //Xay dung hanh vi di chuyen cua nhan vien
        //ghi de hoan toan nen khong co super
    }

    @Override
    Void inTT(){
        //Goi den phuong thuc inTT cua doi tuong cha
        Super.inTT();
        //in them thong tin cho cac thuoc tinh ma NV, chuc vu, luong
    }
}

```

❖ Đối tượng Quản Lý

```
Class QuanLy{
//khai bao danh sach nhan vien
    NhanVien[] dsNhanVien;
//khai bao danh sach hoc sinh
    HocSinh[] dsHocSinh;

    Void themNhanVien(NhanVien nhanvien){
//them nhan vien vao danh sach
    }
    Void themHocSinh(HocSinh hocsinh){
//them hoc sinh vao danh sach
    }
    NhanVien timKiemNhanVien( String maNV){
//tim kiem va tra ve thong tin nhan vien theo ma nhan vien
    Return null;
//Tra ve gia tri mac dinh
    }
    HocSinh timKiemHocSinh( String maHS){
//tim kiem va tra ve thong tin hoc sinh theo ma học sinh
    Return null;
//Tra ve gia tri mac dinh
    }
    Void xoaNhanVien(String maNV){
//Xoa nhan vien theo ma nhan vien
    }
    Void xoaHocSinh( String maHS){
//Xoa hoc sinh theo ma hoc sinh
    }
    Void hienThiDSHS(){
//Hien thi thong tin cua tat ca hoc sinh
    }
    Void hienThiDSNV(){
//Hien thi thong tin cua tat ca nhan vien
    }
}
```

IV: TÍNH ĐÓNG GÓI VÀ CHE DẤU DỮ LIỆU

1. Tính đóng gói

- Là việc phân chia các đối tượng có cùng chung bản chất về cùng một nhóm để quản lý.
- Tính đóng gói giúp kiến trúc chương trình trở nên mạch lạc, tối ưu, việc bảo trì, sửa chữa và nâng cấp trở nên dễ dàng hơn.

2. Cách xây dựng các gói(package) chương trình

- Mỗi chương trình sẽ có một gói chính để chứa toàn bộ mã nguồn của chương trình.
- Cách đặt tên gói chính được quy định như sau:

Com.tên_tổ_chức.tên_project

quy tắc đặt tên:

- **com** viết tắt của company
- Chỉ sử dụng các chữ cái a-z, 0-9 để đặt tên cho các gói chương trình.
- **tên_tổ_chức** ví dụ như: samsung,...hoặc tên biệt danh của một cá nhân nào đó.
- Sau khi xây dựng gói chính, chúng ta tiến hành gom nhóm các đối tượng có cùng 1 bản chất vào 1 nhóm và đặt tên nhóm cho chúng. Như vậy các gói riêng biệt sẽ nằm trong gói chính. Mỗi gói có ít nhất 1 đối tượng.
- Khai báo: **package tên_gói** ở trên đầu mỗi đối tượng.

Ví dụ phân tích: Hãy in thông tin chu vi, diện tích, độ dài cạnh, chiều dài, chiều rộng của hình vuông và hình chữ nhật nếu biết thông tin 4 đỉnh của hình vuông, hình chữ nhật đó.

- Các đối tượng có mặt trong bài toán: Hình Vuông, Hình Chữ Nhật, Tứ Giác, Đỉnh.
- Áp dụng tính đóng gói ta có như sau:

- Gói chính: **com.na.tthcnhv**
- Phân chia các đối tượng vào các gói riêng biệt như sau:
 - Gói **com.na.tthcnhv.dagiac** gồm các đối tượng sau:
 - TuGiac
 - Gói **com.na.tthcnhv.tugiac** gồm các đối tượng sau:
 - TuGiac
 - HinhVuong
 - HinhChuNhat
 - Gói **com.na.tthcnhv.dinh** gồm các đối tượng sau:
 - Dinh
 - Sau này, nếu bài toán có thể phát triển mở rộng, có thêm:
 - Đối tượng Hình Bình Hành kế thừa Tứ Giác thì chúng ta sẽ thêm Hinh Binh Hanh vào gói **tugiac**.
 - Đối tượng Tam Giác Cân, Tam Giác Vuông kế thừa Tam Giác thì chúng ta sẽ thêm:

TamGiac vào gói **dagiac**

TamGiacCan, TamGiacVuong vào gói **tamgiac**(gói này sẽ cần được tạo thêm).

Code biểu diễn:

❖ Đối tượng Tứ Giác

```
Package com.na.tthcnhv.dagiac;
Calss TuGiac{
....
```

❖ Đối tượng hình vuông

```
Package com.na.tthcnhv.tugiac;  
Class HìnhVuong extends TuGiac{  
...  
}
```

❖ Đối tượng Đỉnh

```
Package com.na.tthcnhv.dinh;  
Class Dinh{  
...  
}
```

❖ Đối tượng Hình Chữ Nhật

```
Package com.na.tthcnhv.tugiac;  
Class HìnhChuNhat extends TuGiac{  
...  
}
```

❖ Đối tượng Tam Giác

```
Package com.na.tthcnhv.dagiac;  
Class TamGiac{  
...  
}
```

3. Tính che dấu dữ liệu

- Che dấu dữ liệu là việc quy định phạm vi chia sẻ dữ liệu bao gồm (thuộc tính, phương thức) của một đối tượng.

- Đối tượng A sẽ được khai báo quyền riêng tư cho các thuộc tính và phương thức của mình, qua đó, các đối tượng khác sẽ có các mức độ truy cập và sử dụng thông tin của đối tượng A là khác nhau.
 - **Ví dụ:** Một ông bố có một chiếc xe máy và 1 hũ vàng là tài sản, ông quy định chiếc xe máy thì ai cũng có thể sử dụng được, ai cũng có thể mượn. Nhưng còn hũ vàng thì ông chỉ để dành cho các con của ông.
- **Kết luận:** Như vậy thì chúng ta coi ông bố là đối tượng A, có 2 thuộc tính là *xe máy* và *hũ vàng*, ông bố đã định nghĩa quyền riêng tư cho các thuộc tính của mình, do đó bạn bè, hàng xóm đóng vai trò là các đối tượng trong mối quan hệ phụ thuộc hoặc song song đều có thể sử dụng được thuộc tính xe máy. Còn hũ vàng thì chỉ có các đối tượng là con của ông trong mối quan hệ kế thừa sử dụng được.

4. Tác dụng của tính che dấu dữ liệu

- Bảo vệ tính toàn vẹn dữ liệu của một đối tượng
 - Một đối tượng B sẽ có tối đa 2 quyền để sử dụng dữ liệu của đối tượng A đó là: Quyền đọc và quyền ghi.
 - Do đó, che giấu dữ liệu giúp đối tượng A bảo vệ được dữ liệu của mình không bị đối tượng khác can thiệp vào để sửa đổi.
- Làm nổi bật vai trò của một đối tượng đối với các đối tượng khác trong các ngữ cảnh sau:
 - Trong cùng một lớp: Nghĩa là nội tại bên trong đối tượng đó.
 - Trong cùng một gói
 - Khác gói
 - Mối quan hệ kế thừa

- Ví dụ: Một người có các thuộc tính như: tên, tuổi, mối quan hệ, tên vợ, tên con, tài khoản tiết kiệm cho con, có các hành vi sau: giảng dạy, ca hát, cho con ăn, dạy con học, đi chơi với bạn, đá bóng, đầu tư cho con học tập. Do đó:
 - Nếu là trong cùng một lớp (Có nghĩa là bản thân đối tượng này) sẽ hiểu hết tất cả vì chính là bản thân anh ta.
 - Nếu trong cùng 1 gói **giadinh** thì các đối tượng khác như vợ và con của đối tượng này sẽ đều biết một số thuộc tính: tên, tên vợ, tên con, tuổi. Và hành vi: ca hát, cho con ăn, dạy con học.
 - Nếu khác gói **giadinh** thì những người bạn của đối tượng này sẽ biết đến một số thuộc tính như: tên, và một số hành vi như: giảng dạy, đi chơi với bạn, đá bóng.
 - Nếu trong mối quan hệ kế thừa, đối tượng con có thể biết được thuộc tính: Tài khoản tiết kiệm cho con và các hành vi như: Đầu tư cho con học tập.

5. Cách định nghĩa quyền riêng tư cho đối tượng

5.1. Cú pháp khai báo quyền riêng tư

- Khai báo quyền riêng tư cho class, thuộc tính, phương thức
 - **Mức_độ_r riêng_tư** class TenLop{...
 - **Mức_độ_r riêng_tư** <kiểu dữ liệu> tenThuocTinh;
 - **Mức_độ_r riêng_tư** <kiểu dữ liệu> tenPhuongThuc(...;

5.2. Bảng ý nghĩa quyền riêng tư và phạm vi truy cập

	Phạm vi	Internal class	Trong cùng gói	MQH kế thừa	Khác gói
Quyền					
<i>private</i>		O	X	X	X
<i>default</i>		O	O	X	X
<i>protected</i>		O	O	O	X
<i>public</i>		O	O	O	O

Giải thích:

- Đối tượng A nằm trong gói **com.na.a**, có 4 thuộc tính: a1,a2,a3,a4 và khai báo quyền riêng tư như sau:
 - a1(private)
 - a2 (default)
 - a3 (protected)
 - a4 (public)
- **Internal class:** Là phạm vi truy cập mức đối tượng: Có nghĩa là việc truy cập và sử dụng các thuộc tính của đối tượng A là xảy ra bên trong đối tượng A. Do đó, đối tượng A có thể sử dụng được các thuộc tính a1, a2, a3, a4 mà không cần quan tâm đến việc khai báo quyền riêng tư.
- **Trong cùng gói:** Là phạm vi truy cập mức gói(package): Có nghĩa là việc truy cập và sử dụng các thuộc tính của đối tượng A là xảy ra bên trong gói **com.na.a**. Giả sử có thêm đối tượng B trong gói **com.na.a**. Lúc đó đối tượng B:
 - Không thể truy cập đến thuộc tính của a1 và a1 khai báo quyền riêng tư là **private**(chỉ dành cho truy cập mức đối tượng).
 - Có thể truy cập đến các thuộc tính a2, a3, a4. Hay nói cách khác (default, protected, public) là 3 quyền riêng tư dành cho mức gói(package).

- **Khác gói:** Là phạm vi truy cập mức khác gói(diff package): Có nghĩa là việc truy cập và sử dụng các thuộc tính của đối tượng A là xảy ra bên ngoài gói **ccom.na.a**. Lúc đó, đối tượng C:
 - Không thể truy cập đến thuộc tính a1 vì a1 khai báo quyền riêng tư là **private**
 - Không thể truy cập đến a2 và a3. Hay nói cách khác(**default và protected**) là 2 quyền riêng tư chỉ dành cho mức gói(**package**).
 - Có thể truy cập đến thuộc tính của a4. Hay nói cách khác (**public**) là quyền riêng tư dành cho truy cập mức khác gói (diff package).
- **MQH kế thừa:** là phạm vi truy cập trong MQH kế thừa(inherited): Có nghĩa là việc truy cập và sử dụng các thuộc tính của đối tượng A xuất phát từ các đối tượng con. Giả sử có đối tượng A1 kế thừa đối tượng A. Lúc này A 1 chỉ có thể nằm trong cùng gói hoặc khác gói với đối tượng A, Đối tượng A1:
 - Không thể truy cập đến thuộc tính a1 vì a1 khai báo quyền riêng tư là **private**
 - Không thể truy cập đến thuộc tính a2 vì a2 khai báo quyền riêng tư là **default** - là quyền riêng tư chỉ dành cho mức gói(package).
 - Có thể truy cập đến thuộc tính a3 và a4. Hay nói cách khác (**protected và public**) là quyền riêng tư dành cho truy cập trong MQH kế thừa.

5.3. Quy tắc khai báo quyền riêng tư cho các thuộc tính, phương thức của đối tượng.

- **Đối với các thuộc tính:**
 - Chỉ được khai báo ở hai mức:
 - **private** nếu thuộc tính đó chỉ của riêng đối tượng.

- **protected** nếu thuộc tính đó được các đối tượng con kế thừa.

- Ví dụ:

```
class Dad{  
    private String job;  
    private double salary;  
  
    protected String name;  
    protected int age;  
}
```

```
class Child extends Dad{  
    private String className;  
    private int pointMath;  
}
```

- **Đối với các phương thức:**
 - Tùy thuộc vào phạm vi truy cập của các thuộc tính khác để khai báo quyền riêng tư cho hợp lý.
 - Nếu các phương thức chỉ được sử dụng ở trong nội bộ của đối tượng thì khai báo private.
 - Nếu các phương thức chỉ được sử dụng ở trong cùng 1 gói thì khai báo là default(không cần khai báo).
 - Nếu các phương thức chỉ được sử dụng ở trong cùng 1 gói thì khai báo default(không cần khai báo).
 - Nếu các phương thức chỉ được sử dụng ở trong đối tượng con thì khai báo là protected
 - Nếu các phương thức được sử dụng ở các đối tượng ở khác gói thì khai báo public.
- **Đối với các thuộc tính là hằng số:**
 - Có thể khai báo quyền riêng tư là **public static final** khi cần thiết. Vì đối với loại thuộc tính này, các đối tượng

khác chỉ có thể sử dụng chứ không thể thay đổi được giá trị của nó.

- Ví dụ: **public static final float PI = 3.14F;**
- Ý nghĩa của việc khai báo **public static final**: Nó giúp cho các đối tượng khác có thể nhìn thấy và sử dụng ngay được thuộc tính hằng của một đối tượng khác.

6. Vấn đề phát sinh

- Ý nghĩa lớn nhất của việc chỉ được khai báo ở hai mức: private hoặc protected với các thuộc tính là: bảo vệ sự toàn vẹn dữ liệu của một đối tượng, ngăn không cho các đối tượng khác có thể truy cập để thay đổi dữ liệu.
- Nếu đối tượng B truy cập đến thuộc tính của đối tượng A sẽ có hai hình thức:
 - Lấy thông tin thuộc tính của đối tượng A(Hình thức 1)
 - Sửa thông tin thuộc tính của đối tượng A(Hình thức 2)
- **Bài toán đặt ra:** Nếu đối tượng B chỉ muốn truy cập để lấy thông tin thuộc tính của đối tượng A(Hình thức 1) thì phải làm thế nào? Vì nếu khai báo quyền riêng tư cho thuộc tính của đối tượng A là private thì đối tượng B sẽ không thể truy cập được đến thuộc tính của đối tượng A.

➤ Giải pháp:

- Để giải quyết hai vấn đề:
 - Bảo vệ toàn vẹn dữ liệu của một đối tượng, ngăn không cho các đối tượng khác có thể truy cập để thay đổi dữ liệu.
 - Cho phép đối tượng B truy cập để lấy thông tin thuộc tính của đối tượng A.(Hình thức 1)
- Ta sẽ xây dựng thêm phương thức **get** để trả về giá trị của thuộc tính và khai báo quyền riêng tư cho phương thức để đối tượng B có thể truy cập được.

➤ **Cú pháp:**

```
Public <kiểu dữ liệu> get tên_thuộc_tính(){  
    Return tên_thuộc_tính;  
}
```

➤ **Ví dụ:**

```
Class Child extends Dad{  
    Private String className;  
    Private int pointMath;  
  
    Public String getClassName(){  
        Return className;  
    }  
    ....  
}
```

V: TÍNH ĐA HÌNH

1. Định nghĩa

- Là việc triệu gọi đến các phương thức được ghi đè từ đối tượng cha trong các đối tượng con
- Cùng là một phương thức được định nghĩa hoặc xây dựng trong đối tượng cha, tuy nhiên các lớp con khi kế thừa cha sẽ có thể ghi đè lại các hành vi đó. Do vậy mà các hành vi đó, khi các lớp con thực hiện thì sẽ theo các cách khác nhau, điều này được gọi là đa hình.
- Ví dụ

Đối tượng	Hành vi
Tứ Giác	//Định nghĩa hành vi trừu tượng: Tính chu vi Protected abstract void tinhChuVi();
Hình Vuông	@override Public void tinhChuVi(){ chuVi= doDaiCanh * 4; }
Hình Chữ Nhật	@override Public void tinhChuVi(){ chuVi = (chieuDai + chieuRong) * 2; }

2. Ý nghĩa của đa hình

- Quản lý danh sách đối tượng có mối quan hệ kế thừa.
- Quản lý các bộ hành vi chung của một nhóm các đối tượng giống hoặc khác nhau về bản chất.

2.1. Quản lý danh sách đối tượng có mối quan hệ kế thừa

a. Định nghĩa

- Khi phải quản lý danh sách hai hoặc nhiều loại đối tượng có mối quan hệ kế thừa, chúng ta quy hết về quản lý danh sách duy nhất của một loại đối tượng, đó là lớp cha của nhóm các đối tượng đó.
- Ví dụ: Quản lý danh sách hình chữ nhật và hình vuông => Quy về quản lý lớp cha của hai đối tượng trên đó là đối tượng Tứ Giác

Tách riêng từng loại đối tượng	Quy về một đối tượng
HìnhChuNhat hcn = new HìnhChuNhat(); HìnhVuong hv = new HìnhVuong(); hcn.tinhCV(); hv.tinhCV(); hcn.tinhDT(); hv.tinhDT(); hcn.inTT(); hv.inTT();	TuGiac tg1 = new HìnhChuNhat(); TuGiac tg2 = new HìnhVuong(); tg1.tinhCV(); tg2.tinhCV(); tg1.tinhDT(); tg2.tinhDT(); tg1.inTT(); tg2.inTT();

- Như vậy: Thay vì quản lý từng loại đối tượng, chúng ta quy về một loại đối tượng duy nhất là Tứ Giác, qua đó, chúng ta sẽ không cần phải bận tâm các đối tượng trong danh sách cụ thể là các đối tượng nào, chỉ cần gọi các hành vi chung của lớp Tứ Giác, khi đó, từng phần tử con trong danh sách sẽ tự biết thể hiện cho đúng bản chất của mình:


```
tg1.tinhCV(); //Tứ giác 1 sẽ thực hiện hành vi tính chu vi Hình Chữ Nhật  
tg2.tinhCV(); //Tứ giác 2 sẽ thực hiện hành vi tính chu vi Hình Vuông
```

b. Từ khóa “instanceOf”

- Được sử dụng để kiểm tra kiểu dữ liệu của một đối tượng có phải là kiểu dữ liệu đang nghi ngờ hay không. Nếu đúng trả về True, còn sai thì trả về False
- Ví dụ: `TuGiac tg1 = new HìnhChuNhat();`
⇒ **Boolean** `kt = tg1 instanceof HìnhChuNhat;`
⇒ `Kt = true;`

c. Vấn đề phát sinh

- Nếu sử dụng khai báo: `TuGiac tg1 = new HìnhChuNhat();` thì chỉ có thể gọi đến các hành vi chung được khai báo trong đối tượng Tứ Giác, không thể gọi đến các hành vi riêng của đối tượng Hình Chữ Nhật như `inTTCD CR()`. Vậy thì phải làm thế nào?

Bài Toán: In thông tin chiều dài, chiều rộng của tất cả các hình chữ nhật trong danh sách.

Cách giải quyết:

- B1: kiểm tra kiểu dữ liệu của phần tử thứ (i) trong danh sách xem có phải là kiểu dữ liệu cần chuyển đổi hay không? Trong trường hợp này, kiểu dữ liệu cần chuyển đổi là `HìnhChuNhat`.
- B2: Nếu B1 cho kết quả là (true) thì ép kiểu dữ liệu của phần tử thứ (i) đó về kiểu dữ liệu cần chuyển đổi. Sau đó gọi đến các hành vi mong muốn

```
if( tg1 instanceof HìnhChuNhat){  
    HìnhChuNhat hcn = (HìnhChuNhat)tg1;  
    Hcn.inTTCDCCR;  
}
```

2.2. Quản lý bộ hành vi của các đối tượng không cùng bản chất

a. Định nghĩa

- Khi phải quản lý một hoặc một nhóm các hành vi của danh sách hai hoặc nhiều loại đối tượng không cùng bản chất, chúng ta xây dựng một bộ hành vi chung cho các đối tượng và quản lý chúng dựa trên bộ hành vi đó.

Bài toán: Quản lý hành vi tính chu vi, diện tích của các đối tượng hình tròn, hình vuông, hình chữ nhật.

Phân tích:

- 3 đối tượng **Hình Tròn, Hình Vuông, Hình Chữ Nhật** không cùng bản chất nên không thể áp dụng các phương thức quy về đối tượng cha để xử lý.
- Xây dựng một bộ hành vi chung: tính chu vi, tính diện tích
- Cho 3 đối tượng trên thực hiện bộ hành vi chung để có cơ sở quản lý.

b. Từ khóa “interface”

- Được sử dụng để định nghĩa bộ hành vi cho đối tượng

Cú pháp:

<pre>interface Ten_Bo_Hanh_Vi{ //Định nghĩa các hành vi dung chung }</pre>	<pre>interface BoHanhViCVDT{ void tinhChuVi(); void tinhDienTich(); }</pre>
--	---

c. Từ khóa “implement”

- Được sử dụng khi một đối tượng muốn thực thi một bộ hành vi đã được định nghĩa
- Một đối tượng có thể thực thi nhiều bộ hành vi khác nhau.

Cú pháp:

```
Class Tên_Đối_Tượng implements BoHanhVi1, BoHanhVi2,...{  
    //ghi đè các hành vi của BoHanhVi1, BoHanhVi2,...  
}
```

Lưu ý:

- Một đối tượng chỉ có thể thừa kế một đối tượng khác, tuy nhiên lại có thể thực thi nhiều bộ hành vi.
- Một đối tượng vừa có thể kế thừa một đối tượng vừa có thể thực thi các bộ hành vi. Trong trường hợp này cần biểu diễn mối quan hệ kế thừa trước, sau đó mới đến thực thi bộ hành vi.

PHẦN B: DỰ ÁN

DỰ ÁN QUẢN LÝ LƯƠNG NHÂN VIÊN CỦA MỘT CÔNG TY

I. GIỚI THIỆU DỰ ÁN:

1. Mục đích

- Trong bất kỳ quốc gia nào, ở các doanh nghiệp, các tổ chức kinh tế...kế toán bao giờ cũng có một vị trí đặc biệt quan trọng. Công tác kế toán ở các doanh nghiệp Việt Nam là một bộ phận không thể thiếu ở tất cả các đơn vị tổ chức. Trong đó kế toán tiền lương nhân viên là một bộ phận quan trọng trong doanh nghiệp, tổ chức kinh tế. Nó góp phần quyết định sự tồn tại và phát triển của doanh nghiệp và tổ chức kinh tế đó.

- Quản lý lương là một công việc phức tạp, đòi hỏi sự tỉ mỉ, chi tiết và độ chính xác cao. Quản lý lương một cách có hiệu quả không đơn giản nếu chỉ sử dụng những phương pháp quản lý lương truyền thống như trước đây.

- Đối với một doanh nghiệp, tổ chức kinh tế thì việc xây dựng được một hệ thống quản lý lương sao cho có hiệu quả, đảm bảo sự công bằng, chính xác, thuận tiện là một trong những công tác được đặt lên hàng đầu nhằm ổn định nhân lực, phục vụ cho hoạt động của doanh nghiệp, tổ chức kinh tế.

2. Thực tế hiện nay

- Ngày nay, với sự phát triển của khoa học kỹ thuật thì việc áp dụng công nghệ thông tin vào công tác quản lý đã không còn xa lạ nữa. Do tính chất quan trọng của việc quản lý tiền lương, một loạt các phần mềm quản lý lương, các dạng máy chấm công được ra đời. Tuy nhiên, giá thành của các sản phẩm này là không nhỏ và để áp dụng cho một tổ

chức kinh doanh nào đó thì đôi khi lại không đáp ứng được các nhu cầu cụ thể hoặc các chức năng của chúng quá rộng lớn mà tổ chức kinh doanh đó không sử dụng hết thì rất lãng phí.

- Xuất phát từ vấn đề đó nên nhóm chúng em xin chọn đề tài “**Phân tích thiết kế Dự án Quản lý lương nhân viên** “. Nhóm chúng em nghiên cứu đề tài này với mong muốn phục vụ cho việc quản lý lương của các doanh nghiệp, tổ chức kinh tế.. được thuận tiện hơn và tránh những sai sót trong công tác quản lý tiền lương.

3. Hệ thống được đề xuất

- Dự án xây dựng sẽ mang lại sẽ mang lại nhiều lợi ích cho công ty, doanh nghiệp các tổ chức kinh tế. Ngoài việc có thể quản lý được tiền lương Phần mềm còn có thể quản lý thông tin nhân viên, chấm công nhân viên một cách chính xác, giúp cho doanh nghiệp, công ty, tổ chức kinh tế tiết kiệm và hiệu quả trong công tác quản lý lương và quản lý nhân viên...

II. MÔ TẢ TỔNG THỂ

1. Người sử dụng hệ thống

- Quản trị viên
- Người dùng:
 - + Quản lý nhân viên
 - + Nhân viên Kế Toán

2. Hệ thống chức năng

- Quản lý nhân viên: Xem danh sách nhân viên, cập nhật, thêm sửa xóa, thông tin chi tiết từng nhân viên.
- Nhân viên Kế Toán: Cập nhật thêm sửa xóa các thông tin về lương như hệ số lương, tiền thưởng, phụ cấp, cập nhật chấm công nhân viên, xem danh sách, in bảng lương...

- Quản trị viên: Theo dõi thống kê chi tiết nhân viên, lương, chấm công, thêm xóa người dung.

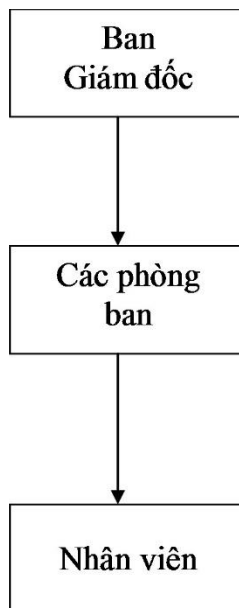
III. PHÂN TÍCH YÊU CẦU

1. Thông tin và yêu cầu của khách hàng:

- Đây là bản mẫu mà doanh nghiệp đặt hàng thiết kế để phục vụ cho công tác quản lý tính lương trong doanh nghiệp. Với các yêu cầu cụ thể của doanh nghiệp như: Thêm mới và cập nhật thông tin nhân viên, phòng ban, quản lý hệ số lương, phụ cấp, tiền thưởng, chấm công nhân viên, tính tiền lương cho từng nhân viên, thống kê lương...

2. Thông tin qua khảo sát thực tế:

- Cơ cấu tổ chức và cách tính lương của một công ty:



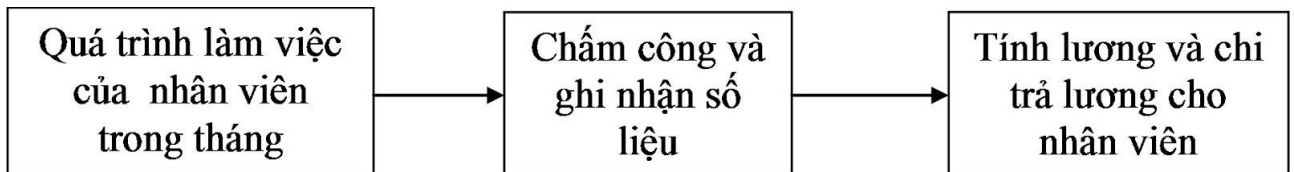
(Sơ đồ cơ cấu tổ chức và cách tính lương của công ty)

- Trong công ty việc quản lý các nhân viên được chia thành các phòng ban để tiện cho việc quản lý các hoạt động cũng như quá trình làm việc của nhân viên.

3. Mô tả hệ thống phần mềm tính lương

Quy trình tính lương của công ty:

- thông thường việc tính lương nhân viên được thực hiện theo quy trình sau.



- Quy trình tính lương của nhân viên công ty được thực hiện sau mỗi tháng làm việc, mỗi tháng nhân viên làm việc sẽ được ghi nhận lại số ngày làm việc và thời gian làm việc để tiến hành tính lương.
- Khi tính tiền lương, mỗi công việc của từng công nhân viên tham gia lao động được tính toán theo các công thức và dựa trên các tiêu chí khác nhau.

a. Cách tính lương:

Công ty quy định như sau:

* *Lương cơ bản 1 ngày làm việc 8h (LCB)=100.000đ (2.600.000đ/tháng)*

* *Lương cho mỗi ngày làm thêm = 150.000đ*

* *Lương cho mỗi giờ thêm= 15.000đ*

* *Hệ số lương: HSL*

Ø Lương của công nhân viên

- Số tiền lương trong một tháng mà mỗi công nhân nhà máy nhận được sẽ được tính theo công thức sau: (bắt buộc phải làm trên hoặc bằng 26 ngày, công nhân viên làm dưới 26 ngày sẽ bị trừ tiền thưởng).

Tiền lương = (26*LCB*HSL) + Lương tăng ca+ Phụ cấp + Tiền

thưởng -Tiền ứng trước – BHXH

Ø Lương tăng ca của công nhân viên

- Lương tăng ca của công nhân viên sẽ được tính toán theo lương tăng ca theo giờ và lương tăng ca theo ngày.

*** Lương tăng ca theo giờ = Số giờ tăng ca x tiền tăng ca 1 giờ(15.000VND/giờ) x HSL**

*** Lương tăng ca theo ngày = Số ngày tăng ca x 1.5 x Tiền lương ngày (100.000VND/Ngày)x Hệ số lương.**

4. Đặc tả bài toán tính lương nhân viên

- Quản lý thông tin chấm công của công nhân viên hàng tháng: được quản lý dựa vào một số thông tin và thuộc tính quan trọng như: số ngày làm việc, số giờ làm thêm ,tiền thưởng, tiền phụ cấp, tiền ứng trước, hệ số lương.

- Ngoài ra chúng ta còn thực hiện công tác quản lý tính lương nhân viên theo phòng ban, để khi tiến hành tính lương hoàn tất chúng ta sẽ chi trả cho nhân viên theo từng phòng ban nhằm tạo ra việc chi trả lương được chặt chẽ và nhanh chóng hơn.

- Quản lý tính lương tùy thuộc vào quá trình làm việc của công nhân viên trong tháng, bởi lẽ sau mỗi tháng làm việc ta sẽ cập nhật lại thông tin về quá trình làm việc của nhân viên để từ đó chương trình sẽ tự động tính lương và in bản lương cho nhân viên theo yêu cầu cụ thể.

- Quy trình về tính lương của từng công ty khác nhau, do đó chương trình phần mềm phải đạt yêu cầu là dễ chỉnh sửa và thay đổi theo hoàn cảnh công ty.

IV. GIẢI PHÁP

1. Lựa chọn ngôn ngữ lập trình: *Ngôn ngữ lập trình Java*

- **Java** là một ngôn ngữ lập trình hướng đối tượng (OOP). Khác với phần lớn các ngôn ngữ lập trình thông thường, Java thường chạy nhanh hơn. Ngoài ra Java là một ngôn ngữ lập trình vừa biên dịch vừa thông dịch, viết một lần chạy mọi nơi. Hướng đối tượng trong Java tương tự như C++ đồng thời Java hỗ trợ lập trình đa nhiệm, đa luồng, hỗ trợ mạnh cho việc phát triển ứng dụng. Đây chính là ưu điểm của Java khiến nhóm chúng em chọn ngôn ngữ lập trình này!

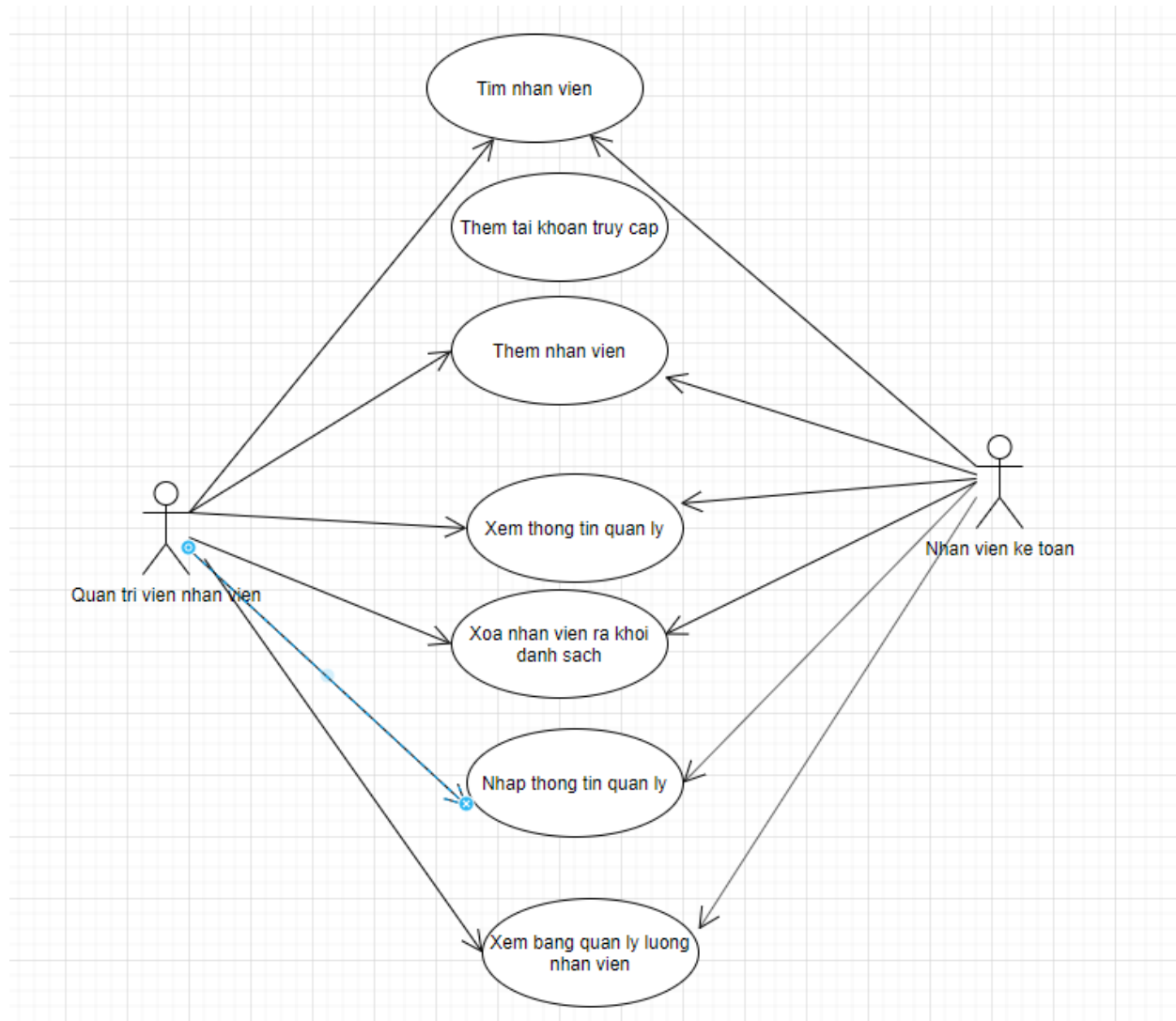
2. Bảng tiến độ và phân công công việc:

Thời gian	Đặng Văn Đức	Hà Mộng Khang
09:40-20/10/2020	Bắt đầu dự án	Bắt đầu dự án
10:00 20/10/2020 đến 12:00 20/10/2020 17:30 20/10/2020 đến 20:00 20/10/2020	Thiết lập các nghiệm vụ cần làm <ul style="list-style-type: none">- Chọn ý tưởng cho dự án.- Xác định trong dự án có những chức năng và nhiệm vụ cụ thể.- Tạo các bài mail lưu thông tin: báo cáo, nguồn thông tin, sản phẩm dự án, quá trình làm việc và kết quả xây dựng theo từng giai đoạn.- Tìm kiếm các nguồn dữ liệu	Thiết lập các nghiệm vụ cần làm <ul style="list-style-type: none">- Chọn ý tưởng cho dự án.- Xác định trong dự án có những chức năng và nhiệm vụ cụ thể.- Tìm kiếm các nguồn dữ liệu phục vụ.- Khảo sát những đối tượng liên quan đến dự án.- Khảo sát những đối tượng liên quan đến dự

	phục vụ	án.
10:00 23/10/2020 đến 12:00 23/10/2020	Xây dựng trang bìa báo cáo dự án quản lý lương nhân viên	Xây các đối tượng cần có trong dự án quản lý lương nhân viên
08:00 26/10/2020 đến 15:30 26/10/2020	Xây dựng báo cáo phân lý thuyết của lập trình hướng đối tượng	Xây dựng sơ đồ diagram quản lý lương nhân viên
08:00 27/10/2020 đến 12:00 27/10/2020	Xây dựng báo cáo phân lý thuyết của lập trình hướng đối tượng	Xây dựng sơ đồ Use case
12:00 31/10/2020 đến 15:30 31/10/2020	Xây dựng báo cáo phân lý thuyết của lập trình hướng đối tượng	Xây dựng code cho dự án quản lý lương nhân viên
12:00 02/11/2020 đến 17:30 02/11/2020	Xây dựng báo cáo phân lý thuyết của lập trình hướng đối tượng	Xây dựng code cho dự án quản lý lương nhân viên
06:18 03/11/2020 đến 19:04 03/11/2020	Xây dựng báo cáo phân lý thuyết của lập trình hướng đối tượng	Xây dựng code cho dự án quản lý lương nhân viên
17:30 04/11/2020 đến 22:05 04/11/2020	Xây dựng báo cáo phân quản lý lương nhân viên	Xây dựng code cho dự án quản lý lương nhân viên
06:28 05/11/2020 đến 09:45 05/11/2020	Xây dựng báo cáo phân quản lý lương nhân viên	Xây dựng báo cáo phân quản lý lương nhân viên

3. Mô hình xây dựng:

a. Sơ đồ use case:



Sơ đồ use case

Mô tả sơ đồ use case:

✖ Use case ID	UC1 ✖
Use case name	Thêm tài khoản truy cập
Description	Quản trị viên, nhân viên đăng nhập tài khoản administrator, sau đó thiết lập tài khoản mật khẩu mới
Pre Condition	Username, password của administrator phải đúng Nhập lại đúng mật khẩu của tài khoản mới Tài khoản không được trùng với tài khoản trước đó
Standard flow	1. Nhập username của admin 2. Nhập password của admin 3. Nhập username password mới 4. Nhập lại password mới
Post condititons	Thiết lập một tài khoản mới, trả về menu đăng nhập

Thêm tài khoản 1

Use case ID	UC2
Use case name	Nhập thông tin quản lý
Description	Người dùng nhập thông tin của các nhân viên
Pre Condition	Đăng nhập thành công
Standard flow	1. Chọn phòng ban 2. Nhập các thông tin cho nhân viên 3. Lựa chọn nhập tiếp hay không
Post condititons	Nhập xong thông tin cho nhân viên trả về menu chính

Nhập thông tin 1

Use case ID	UC4
Use case name	Quản lý lương
Description	Người dùng nhập lựa chọn các chức năng xem lương
Pre Condition	Đăng nhập thành công
Standard flow	1. Chọn chức năng xem lương 2. Đưa ra kết quả
Post condititons	Hiển thị kết quả và trả về menu xem lương

Quản lý lương 1

Use case ID	UC3
Use case name	Thêm nhân viên
Description	Người dùng nhập thông tin của các nhân viên mới
Pre Condition	Đăng nhập thành công
Standard flow	<ol style="list-style-type: none"> 1. Chọn phòng ban 2. Nhập các thông tin cho nhân viên 3. Lựa chọn nhập tiếp hay không
Post condititons	Nhập xong thông tin cho nhân viên trả về menu chính

Thêm nhân viên 1

Use case ID	UC5
Use case name	Tìm nhân viên
Description	Người dùng nhập lựa chọn các chức năng tìm nhân viên
Pre Condition	Đăng nhập thành công
Standard flow	<ol style="list-style-type: none"> 1. Chọn chức năng tìm nhân viên 2. Chọn thông tin cần tìm 3. Hiện thị kết quả
Post condititons	Hiện thị kết quả và trả về menu tìm nhân viên

Tìm nhân viên 1

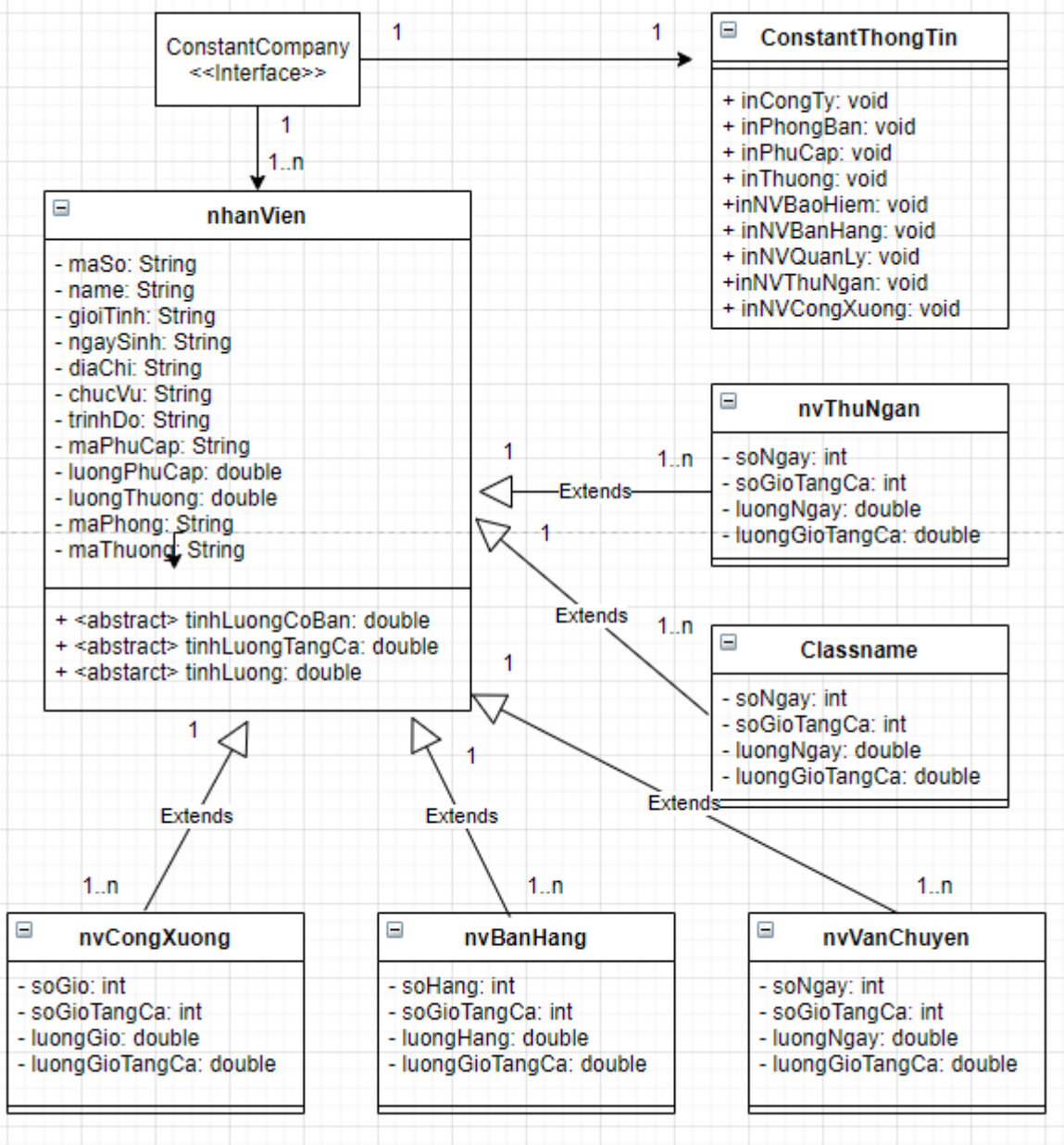
Use case ID	UC6
Use case name	Xóa nhân viên
Description	Người dùng nhập lựa chọn các chức năng xóa nhân viên
Pre Condition	Đăng nhập thành công
Standard flow	1. Nhập mã số nhân viên 2. Xóa nhân viên đó khỏi danh sách
Post condititons	Xóa nhân viên và trả về menu chính

Xóa nhân viên 1

Use case ID	UC7
Use case name	Xem thông tin quản lý
Description	Người dùng nhập lựa chọn các chức năng trong menu xem thông tin
Pre Condition	Đăng nhập thành công
Standard flow	1.Nhập lựa chọn chức năng 2.In kết quả
Post condititons	Hiển thị kết quả và trả về menu xem thông tin

Xem thông tin 1

b. Sơ đồ class diagram:



Sơ đồ class diagram

4: Giao diện ứng dụng:

a. Tài khoản đăng nhập:

- Người quản lý của công ty sẽ được cung cấp một tài khoản Administrator, tài khoản này sẽ được dùng xuyên suốt và chỉ duy nhất tài khoản này có khả năng đăng kí các tài khoản khác và cấp quyền truy cập cho người khác có thể sử dụng các chức năng.
- Tài khoản Admin:
 - + User name: administrator
 - + Password: admin
- Sau khi được người quản lý thêm tài khoản thì tài khoản này sẽ được tồn tại và được đăng nhập thành công, và bạn sẽ không thêm được tài khoản khác trùng với tài khoản đã đăng ký hoặc tài khoản Admin.

1

```
*****
=====LOGIN=====
**          Nhap userName: administrator
**          Nhap password: admin
Ban da dang nhap thanh cong!
.....
```

Giao diện đăng nhập

```
*****
=====LOGIN=====
** Ban phai nhap tai khoan Admin de dang ki tai khoan moi **
**          Nhap userName Admin: administrator
**          Nhap password Admin: admin
*****
=====LOGIN=====
**          Dat userName: khang
**          Dat password: khang2001
**          Nhap lai password: khang2001
Tai khoan da co. Vui long chon lai
```

Giao diện đăng ký tài khoản

Lưu ý:

- Khi bạn nhập lại tài khoản admin sai thì sẽ báo lỗi
- Khi bạn đăng kí nhưng nhập lại mật khẩu sai thì sẽ yêu cầu bạn nhập lại.
- Khi bạn đăng kí tài khoản đã tồn tại thì sẽ thông báo tài khoản đã tồn tại.

b. Giao diện Menu chính:

```
*****
**          CHUONG TRINH QUAN LY LUONG NHAN VIEN          **
*****
=====MENU=====
**          1. Nhap thong tin quan li          **
**          2. Xem thong tin          **
**          3. Them nhan vien          **
**          4. Quan li luong          **
**          5. Tim nhan vien          **
**          6. Xoa nhan vien          **
**          0. Thoat chuong trinh          **
*****
**          Nhap lua chon cua ban          **
```

Giao diện menu chính

- Menu chính bao gồm 7 chức năng:
 - + Nhập thông tin quản lý.
 - + Xem thông tin.
 - + Thêm nhân viên.
 - + Quản lý lương.
 - + Tìm nhân viên.
 - + Xóa nhân viên.
 - + Thoát chương trình.
- Người dùng sẽ nhập các con số tương ứng để có thể sử dụng chức năng nếu người dùng nhập khác các con số qui định thì chương trình sẽ thông báo cho bạn một thông tin bạn nhập không hợp lệ.

c. Giao diện nhập thông tin quản lý:

```
*****
**          CHƯƠNG TRÌNH QUẢN LÝ LƯƠNG NHÂN VIÊN          **
*****
=====MENU=====
**          1. Nhập thông tin quản lý          **
**          2. Xem thông tin          **
**          3. Thêm nhân viên          **
**          4. Quản lý lương          **
**          5. Tìm nhân viên          **
**          6. Xóa nhân viên          **
**          0. Thoát chương trình          **
*****
**          Nhập lựa chọn của bạn          **

1
Ban muốn thêm nhân viên cho phòng ban nào('Cong xuong','Van chuyen','Thu Ngan','Quan ly','Ban hang'):
Cong cong
Ban muốn thêm nhân viên cho phòng ban nào('Cong xuong','Van chuyen','Thu Ngan','Quan ly','Ban hang'):
Cong xuong

*****
**          BẮT ĐẦU NHẬP THÔNG TIN NHÂN VIÊN          **
*****

Nhập mã số nhân viên:
001
Nhập tên:
Ha Mong Khang
Nhập giới tính('nam' hoặc 'nu'):
nam
Nhập ngày sinh:
20/01/2001
Nhập địa chỉ:
Tay Son, Binh Dinh
Nhập chức vụ:
San xuất bao bì
Nhập trình độ:
Cao đẳng
Nhập mã số lương thưởng('cc1','cc2','cc3','tet','tt1','tt2'):
cc1
Nhập mã số phụ cấp('Xa nha','Bao','Xang','An trua','Hoc tap','Them'):
Bao
Nhập số giờ làm:
240
Nhập lương giờ làm:
20000
Nhập số giờ tăng ca:
100
Nhập lương giờ tăng ca:
30000
Ban có muốn nhập tiếp không('yes' hoặc 'no'):
yes
```

Giao diện nhập thông tin quản lý

- Trước tiên bạn phải chọn phòng ban mà mình muốn nhập thông tin, sau khi nhập sẽ có thông báo hỏi bạn muốn nhập cho phòng ban khác không. Sẽ có hai lựa chọn cho bạn.
- Sau mỗi lần kết thúc nhập bạn sẽ được hỏi muốn nhập cho nhân viên tiếp theo không và bạn cần đưa ra 1 trong hai lựa chọn đã được đề xuất.

Lưu ý:

- Mã số nhân viên không được trùng lặp nếu bị trùng lặp sẽ bị thông báo lỗi.
- Không được cho phép nhập giới tính ngoài nam và nữ. Nếu nhập khác sẽ được yêu cầu nhập lại.
- Mã thưởng và mã phụ cấp bạn phải nhập theo quy định công ty.

d. Giao diện menu xem thông tin:

```

*****
**          CHUONG TRINH QUAN LY LUONG NHAN VIEN          **
*****
=====MENU=====
**          1. Nhap thong tin quan li                      **
**          2. Xem thong tin                               **
**          3. Them nhan vien                             **
**          4. Quan li luong                               **
**          5. Tim nhan vien                              **
**          6. Xoa nhan vien                              **
**          0. Thoat chuong trinh                          **
*****
**          Nhap lua chon cua ban                          **
*****

*****
**          CHUONG TRINH QUAN LY LUONG NHAN VIEN          **
*****
=====MENU=====
**          1. Xem thong tin cong ty                       **
**          2. Xem danh sach toan bo nhan vien            **
**          3. Xem danh sach phong ban                   **
**          4. Xem chinh sach bao hiem cua cong ty       **
**          5. Xem chinh sach phu cap cong ty            **
**          6. Xem chinh sach thuong them cua cong ty   **
**          0. Thoat chuc nang                           **
*****
**          Nhap lua chon cua ban                          **
*****

```

- Chương trình sẽ cung cấp cho bạn 6 chức năng xem và bạn sẽ lựa chọn.
- Có các thông tin mặc định của công ty, có các thông tin là của các bạn được nhập vào.

e. Thêm nhân viên:

```

*****
**                CHƯƠNG TRÌNH QUẢN LÝ LƯƠNG NHÂN VIÊN                **
*****
=====MENU=====
**                1. Nhập thông tin quản lý                **
**                2. Xem thông tin                **
**                3. Thêm nhân viên                **
**                4. Quản lý lương                **
**                5. Tìm nhân viên                **
**                6. Xóa nhân viên                **
**                0. Thoát chương trình                **
*****
**                Nhập lựa chọn của bạn                **

3
Bạn muốn thêm nhân viên cho phòng ban nào('Công xưởng','Văn chuyên','Thu Ngan','Quản lý','Ban hàng'
Ban hàng

*****
**                BẮT ĐẦU NHẬP THÔNG TIN NHÂN VIÊN                **
*****

Nhập mã số nhân viên:
bh002
Nhập tên:
Nguyễn Thị Ly
Nhập giới tính('nam' hoặc 'nu'):
nu
Nhập ngày sinh:
23/07/1996
Nhập địa chỉ:
Quang Trach, Quang Binh
Nhập chức vụ:
Ban siêu thị
Nhập trình độ:
Đại học
Nhập mã số lương thưởng('cc1','cc2','cc3','tet','tt1','tt2'):
cc2
Nhập mã số phụ cấp('Xa nha','Bao','Xang','Ăn trưa','Học tập','Them'):
Xang
Nhập số sản phẩm làm:
150
Nhập lương sản phẩm:
45000
Nhập số giờ tăng ca:
23
Nhập lương giờ tăng ca:
10000
Bạn có muốn nhập tiếp không('yes' hoặc 'no'):
no

```

f. Giao diện chức năng menu quản lý lương:

```
*****
**          CHƯƠNG TRÌNH QUẢN LÝ LƯƠNG NHÂN VIÊN          **
*****

=====MENU=====
**          1. Nhập thông tin quản lý          **
**          2. Xem thông tin          **
**          3. Thêm nhân viên          **
**          4. Quản lý lương          **
**          5. Tìm nhân viên          **
**          6. Xóa nhân viên          **
**          0. Thoát chương trình          **
*****
**          Nhập lựa chọn của bạn          **

*****
**          CHƯƠNG TRÌNH QUẢN LÝ LƯƠNG NHÂN VIÊN          **
*****

=====MENU=====
**          1. Xem lương nhân viên theo phòng ban          **
**          2. Xem thông tin người có lương cao nhất          **
**          3. Xem thông tin người có lương thấp nhất          **
**          0. Thoát chức năng          **
*****
**          Nhập lựa chọn của bạn          **
```

Giao diện chức năng quản lý lương nhân viên

g. Giao diện menu tìm nhân viên:

```
*****
**          CHƯƠNG TRÌNH QUẢN LÝ LƯƠNG NHÂN VIÊN          **
*****
=====MENU=====
**          1. Tìm nhân viên theo mã Số nhân viên          **
**          2. Tìm nhân viên theo tên                      **
**          3. Tìm nhân viên theo địa chỉ                  **
**          4. Tìm nhân viên có lương lớn hơn nhập vào    **
**          5. Tìm nhân viên có lương bé hơn nhập vào    **
**          0. Thoát chức năng                             **
*****
**          Nhập lựa chọn của bạn                          **
**          **
```

4

Nhập mức lương nhân viên cần tìm:

3000000

Giao diện chức năng tìm nhân viên

PHẦN C: KIẾN THỨC VÀ KINH NGHIỆM ĐẠT ĐƯỢC

1. Hiểu được tư duy lập trình.
2. Nắm được phương pháp phân tích bài toán hướng đối tượng
3. Các khái niệm cơ bản trong lập trình hướng đối tượng
 - a. Đối tượng
 - b. Thuộc tính
 - c. Phương thức
4. Cách xác định đối tượng có mặt trong bài toán
5. Các ví dụ về phân tích bài toán lập trình theo hướng đối tượng
6. Các tính chất của lập trình hướng đối tượng
7. Tính trừu tượng hóa
 - a. Định nghĩa tính trừu tượng hóa
 - b. Phân tích tính trừu tượng hóa
8. Mối quan hệ giữa các đối tượng
 - a. Định nghĩa
 - b. Các mối quan hệ
 - c. Quan hệ kế thừa
 - d. Quan hệ phụ thuộc
 - e. Quan hệ song song
9. Các kiểu dữ liệu trong Java
 - a. Định nghĩa các kiểu dữ liệu
 - b. Kiểu dữ liệu nguyên thủy và kiểu dữ liệu có cấu trúc
10. Cách khai báo một đối tượng, phương thức, thuộc tính
 - a. Khai báo đối tượng Sơ đồ use case
 - b. Khai báo thuộc tính
 - c. Khai báo phương thức
 - d. Cú pháp và quy tắc đặt tên phương thức
11. Tính kế thừa
 - a. Định nghĩa tính kế thừa trong lập trình hướng đối tượng
 - b. Cơ sở hình thành tính kế thừa trong lập trình hướng đối tượng
 - c. Tác dụng tính kế thừa trong lập trình hướng đối tượng

- d. Các ví dụ phân tích tính kế thừa trong lập trình hướng đối tượng
 - e. Đối tượng trừu tượng, và phương thức trừu tượng
12. Tính đóng gói và che dấu dữ liệu
- a. Định nghĩa tính đóng gói dữ liệu
 - b. Cách xây dựng các gói chương trình
 - c. Tính che dấu dữ liệu
 - d. Tác dụng của tính che dấu dữ liệu
 - e. Cách định nghĩa quyền riêng tư cho đối tượng
 - f. Các vấn đề phát sinh trong việc đóng gói và che dấu dữ liệu
13. Khái niệm JDK
14. Khái niệm JVM
15. Toán tử this
- a. Định nghĩa toán tử this trong lập trình hướng đối tượng
 - b. Các ví dụ toán tử this trong lập trình hướng đối tượng
16. Phương thức khởi tạo
- a. Định nghĩa phương thức khởi tạo trong lập trình hướng đối tượng
 - b. Ý nghĩa và ví dụ phân tích hoạt động phương thức khởi tạo trong lập trình hướng đối tượng
17. Tính đa hình trong lập trình hướng đối tượng
- a. Định nghĩa tính đa hình trong lập trình hướng đối tượng
 - b. Ý nghĩa của đa hình
 - Quản lý danh sách đối tượng có mối quan hệ kế thừa
 - Quản lý bộ hành vi của các đối tượng không cùng bản chất
18. Mảng 1 chiều
19. Mảng 2 chiều
20. Các phương thức sắp xếp, so sánh, tìm kiếm....

PHẦN D: KẾT LUẬN

Chân thành cảm ơn bạn đã xem bài viết của chúng tôi!

Tôi hi vọng, với những gì chúng tôi đã tâm huyết trên bài viết này sẽ giúp cho bạn có cái nhìn nhẹ nhàng hơn trong việc học lập trình Công nghệ thông tin.

Danh sách link:

- **Link codes:**
https://drive.google.com/file/d/1YR7HiqUEXwBVMokj_kbddpRCyeAdvlw7/view?usp=sharing
- **Link class diagram:**
<https://drive.google.com/file/d/1LWn1uKxWPdcUSyC-ThyKt4m0xq-uAvuC/view?usp=sharing>