

# DỰ ÁN: HÌNH CHỮ NHẬT

Xây dựng lớp đối tượng quản lý hình chữ nhật và cài đặt các phương thức tính chu vi và diện tích của hình chữ nhật.

- Bước 1: Xác định lớp đối tượng
- Bước 2: Liệt kê tất cả các thuộc tính của đối tượng
- Bước 3: Xây dựng các phương thức get, set cho từng thuộc tính
- Bước 4: Xây dựng các phương thức khởi tạo
- Bước 5: Xây dựng phương thức nhập, xuất
- Bước 6: Xây dựng phương thức xử lý nghiệp vụ tính chu vi (tạo thêm thuộc tính chu vi để lưu trữ)
- Bước 7: Xây dựng phương thức xử lý nghiệp vụ tính diện tích (tạo thêm thuộc tính diện tích để lưu trữ)
- Bước 8: Vào lớp xử lý chính hoặc nơi cần xử lý, tạo ra đối tượng
- Bước 9: Gọi các phương thức xử lý theo yêu cầu bài toán

# DỰ ÁN: QUẢN LÝ SINH VIÊN - VERSION 1 - 1 LỚP ĐỐI TƯỢNG

Xây dựng chương trình cho phép người dùng nhập vào: Tên, Mã SV, điểm Toán, Lý Hóa. Thực hiện các nghiệp vụ dưới đây.

- Tính điểm trung bình từng sinh viên  $(T + L + H)/3$
- Xếp loại từng sinh viên theo:  $\geq 9 \rightarrow$  Xuất Sắc,  $9 < \text{Giỏi} \leq 8$ ,  $8 < \text{Khá} \leq 7$ ,  $< 7 < \text{TB}$  Khá  $\leq 6$ ,  $< 6 < \text{TB} \leq 5$ , còn lại Yếu.

# GIẢI THUẬT: QUẢN LÝ SINH VIÊN - VERSION 1 - 1 LỚP ĐỐI TƯỢNG

- Bước 1: Xác định lớp đối tượng
- Bước 2: Vẽ sơ đồ lớp UML
- Bước 3: Liệt kê tất cả các thuộc tính của đối tượng
- Bước 4: Liệt kê các phương thức đã thấy trước
- Bước 5: Xây dựng lớp đối tượng
- Bước 6: Xây dựng các phương thức get, set cho từng thuộc tính
- Bước 7: Xây dựng các phương thức khởi tạo
- Bước 8: Xây dựng phương thức nhập, xuất
- Bước 9: Xây dựng phương thức xử lý nghiệp vụ tính điểm trung bình (tạo thêm thuộc tính điểm trung bình để lưu trữ)
- Bước 10: Xây dựng phương thức xử lý nghiệp vụ xếp loại (tạo thêm thuộc tính xếp loại để lưu trữ)
- Bước 11: Vào lớp xử lý chính hoặc nơi cần xử lý, tạo ra đối tượng
- Bước 12: Gọi các phương thức xử lý theo yêu cầu bài toán



# MỘT SỐ PHƯƠNG THỨC THƯỜNG XÀI TRONG ARRAYLIST & LINKEDLIST

☑ ArrayList<SinhVien> listSV = new ArrayList<SinhVien>()

☑ listSV.add(sv); // Thêm mới đối tượng vào danh sách

☑ listSV.remove(sv); // Xóa đối tượng khỏi list

☑ listSV.get(i); // Lấy đối tượng tại vị trí i

☑ Duyệt for

```
for (int i = 0; i < listSV.size(); i++) {  
    listSV.get(i); // Lấy sinh viên thứ i  
}
```

- \* Nếu listSV bị null thì sẽ lỗi
- \* Phức tạp khi lấy đối tượng xài
- \* **Lợi:** tìm chỉ số trong danh sách

☑ Duyệt for each

```
for (SinhVien sv : listSV) {  
    // Lấy đối tượng sv và xài  
}
```

- \* Nếu listSV bị null thì bỏ qua ko chạy
- \* Thao tác dễ trên đối tượng xài
- \* **Nhược:** Nếu cần chỉ số phải thêm biến phụ

## DỰ ÁN: QUẢN LÝ SINH VIÊN - VERSION 2 - 2 LỚP ĐỐI TƯỢNG (LỚP DANH SÁCH)

Xây dựng chương trình cho phép người dùng nhập vào: Tên, Mã SV, điểm Toán, Lý Hóa. Thực hiện các nghiệp vụ dưới đây.

- Tính điểm trung bình từng sinh viên  $(T + L + H)/3$
- Xếp loại từng sinh viên theo:  $\geq 9 \rightarrow$  Xuất Sắc,  $9 < \text{Giỏi} \leq 8$ ,  $8 < \text{Khá} \leq 7$ ,  $< 7 < \text{TB Khá} \leq 6$ ,  $< 6 < \text{TB} \leq 5$ , còn lại Yếu.

# DỰ ÁN: QUẢN LÝ SINH VIÊN - VERSION 3 - 3 LỚP ĐỐI TƯỢNG

Xây dựng chương trình THEO HƯỚNG ĐỐI TƯỢNG cho phép người dùng nhập vào: Tên, Mã SV, điểm Toán, Lý Hóa. Cho phép nhập nhiều Sinh viên và thực hiện (Viết hàm tạo dữ liệu giả)

- Tính điểm trung bình từng sinh viên  $(T + L + H)/3$
- Xếp loại từng sinh viên theo:  $\geq 9 \rightarrow$  Xuất Sắc,  $9 < \text{Giỏi} \leq 8$ ,  $8 < \text{Khá} \leq 7$ ,  $7 < \text{TB Khá} \leq 6$ ,  $6 < \text{TB} \leq 5$ , còn lại Yếu. In ra danh sách theo mẫu
- In ra SV có ĐTB cao nhất. In ra danh sách theo mẫu
- In ra tất cả sinh viên Yếu. In ra danh sách theo mẫu
- Tìm sinh viên theo tên. In ra danh sách theo mẫu
- Tìm sinh viên theo mã
- Xóa 1 sinh viên theo mã

# DỰ ÁN: QUẢN LÝ SINH VIÊN - VERSION 3 - 3 LỚP ĐỐI TƯỢNG

STT	Mã SV	Tên SV	Toán	Lý	Hóa	ĐTB	Xếp loại
1	1	Chi	8	8	8	8	Giỏi
2	3	Lan	8	8	8	8	Giỏi
3	4	Cảnh	8	8	8	8	Giỏi
4	6	Hồng	8	8	8	8	Giỏi
5	7	Chi	8	8	8	8	Giỏi
6	8	Nghĩa	8	8	8	8	Giỏi
7	9	Sơn	8	8	8	8	Giỏi
8	12	Hải	8	8	8	8	Giỏi

# XỬ LÝ CÁC CÂU CÒN LẠI

Mã: 001 Tên: An .... ĐTB: 8.3	Mã: 002 Tên: Viên .... ĐTB: 4.1	Mã: 003 Tên: Ngọc .... ĐTB: 9.1	Mã: 004 Tên: Liên .... ĐTB: 3.3	Mã: 005 Tên: Sơn .... ĐTB: 9.1	Mã: 006 Tên: Tuấn .... ĐTB: 8	Mã: 007 Tên: Vũ .... ĐTB: 3.1	Mã: 008 Tên: Long .... ĐTB: 9.1	Mã: 009 Tên: sơn .... ĐTB: 8.1	Mã: 010 Tên: Liên .... ĐTB: 5.7

- Tìm sinh viên theo tên. In ra danh sách theo mẫu
- Tìm sinh viên theo mã
- Xóa 1 sinh viên theo mã



# PHÂN TÍCH MẪU ĐỊNH DẠNG TRƯỚC KHI CODE

STT	Mã SV	Tên SV	Toán	Lý	Hóa	ĐTB	Xếp loại
1	1	Chi	8	8	8	8	Giỏi
2	3	Lan	8	8	8	8	Giỏi
3	4	Cánh	8	8	8	8	Giỏi
4	6	Hồng	8	8	8	8	Giỏi
5	7	Chi	8	8	8	8	Giỏi
6	8	Nghĩa	8	8	8	8	Giỏi
7	9	Sơn	8	8	8	8	Giỏi
8	12	Hải	8	8	8	8	Giỏi

## TruongHoc

```
xuatLine();
xuatHeader();
formatCell(paddLeft, title, paddRight);
xuatCellThuTu(int i);
```

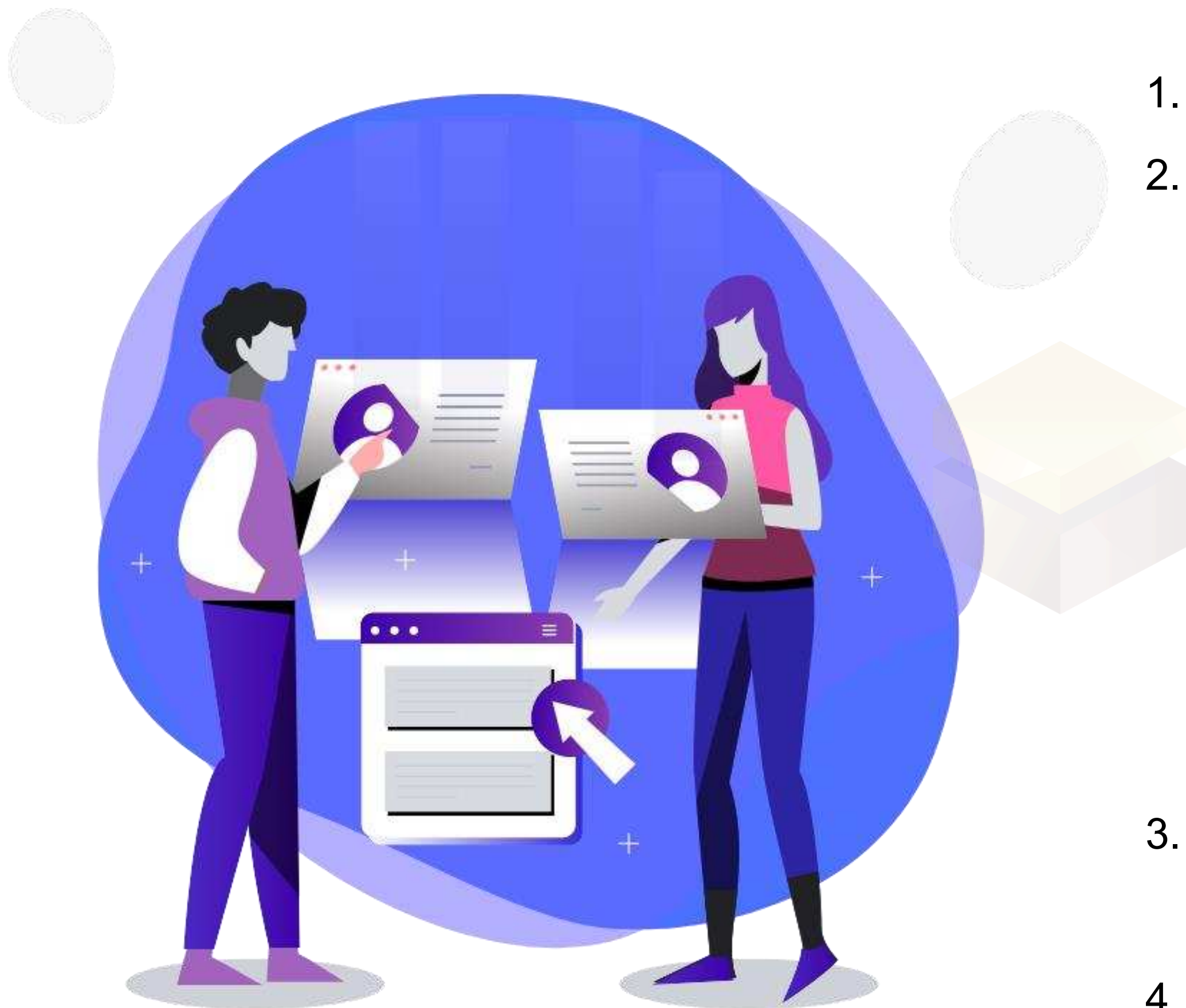
## SinhVien

```
xuatRowFormat();
formatNumCell(Number num);
formatTextCell(String text);
```

# Một số điểm cần lưu ý



1. Phân tích nghiệp vụ → Xác định số lớp đối tượng → Vẽ sơ đồ lớp
2. Tạo các lớp đối tượng, mỗi lớp bao gồm:
  - Danh sách các thuộc tính Attributes.
  - Tập hợp các phương thức Get/Set
  - Tập hợp các phương thức khởi tạo (Cần có ít nhất 1 PT khởi tạo mặc định không tham số)
  - Các phương thức khác
    - Xử lý nhập, xuất
    - Xử lý nghiệp vụ
3. Phương thức nên được cài đặt tại lớp nào ? → Nếu nghiệp vụ cần xử lý liên quan đến một hoặc nhiều thuộc tính của lớp → Đặt trong lớp
4. Ra Lớp xử lý chính, tạo các Instance và gọi thực hiện.

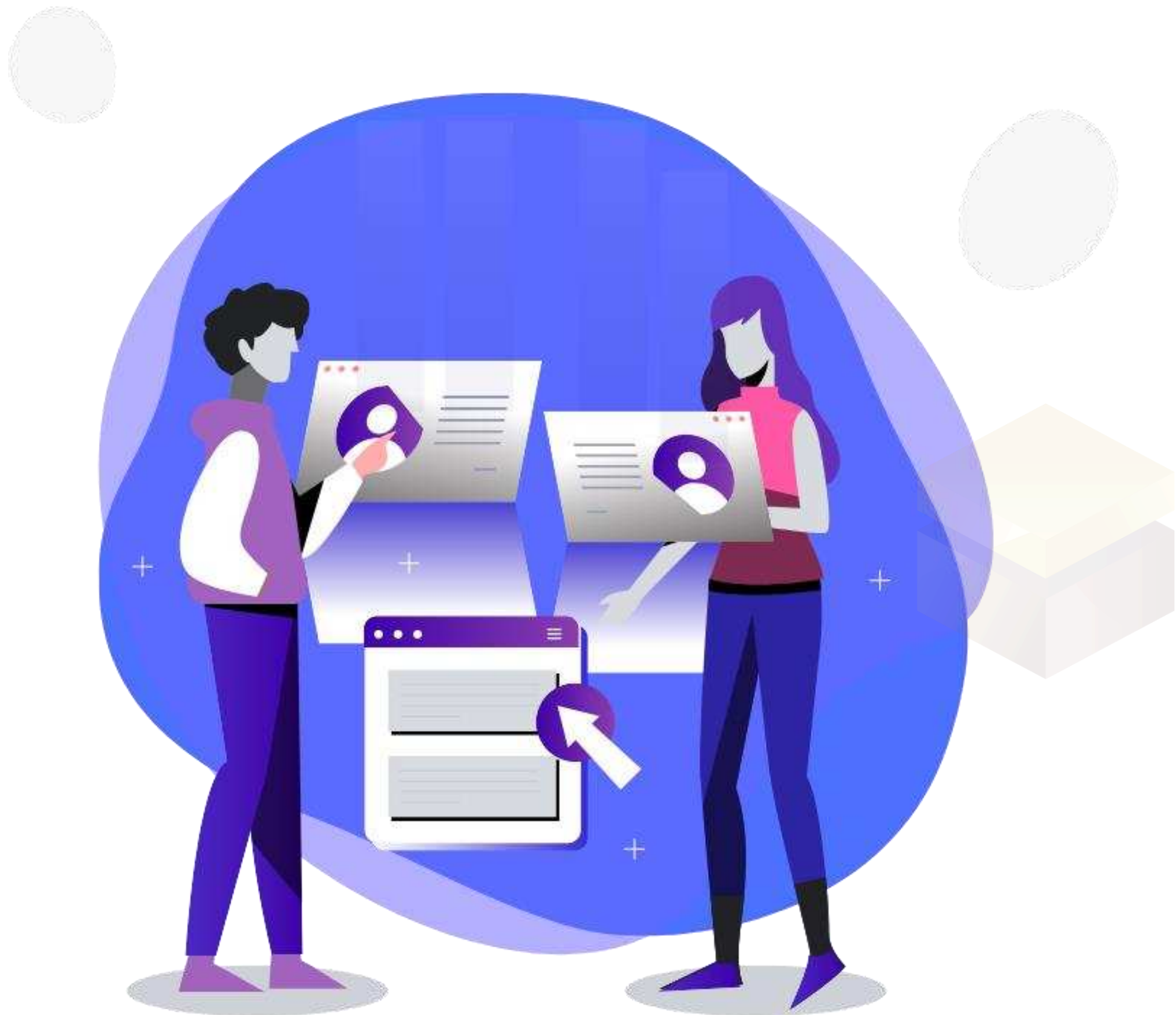


# Một số điểm cần lưu ý (TT)



## Khi làm việc với hướng đối tượng cần nhớ:

- Muốn xài đối tượng thì đối tượng phải được khởi tạo (tạo *instance* hay *new*)
- Chú ý Null Pointer Exception vì đối tượng chưa được khởi tạo mà truy xuất đến các phương thức bên trong
- Tìm giá trị nhiều nhất cao nhất/thấp nhất thì chú ý sẽ có thể có nhiều giá trị cao nhất/thấp nhất cùng giá trị bằng nhau —> Kết quả được danh sách đối tượng chứ không phải là 1 đối tượng





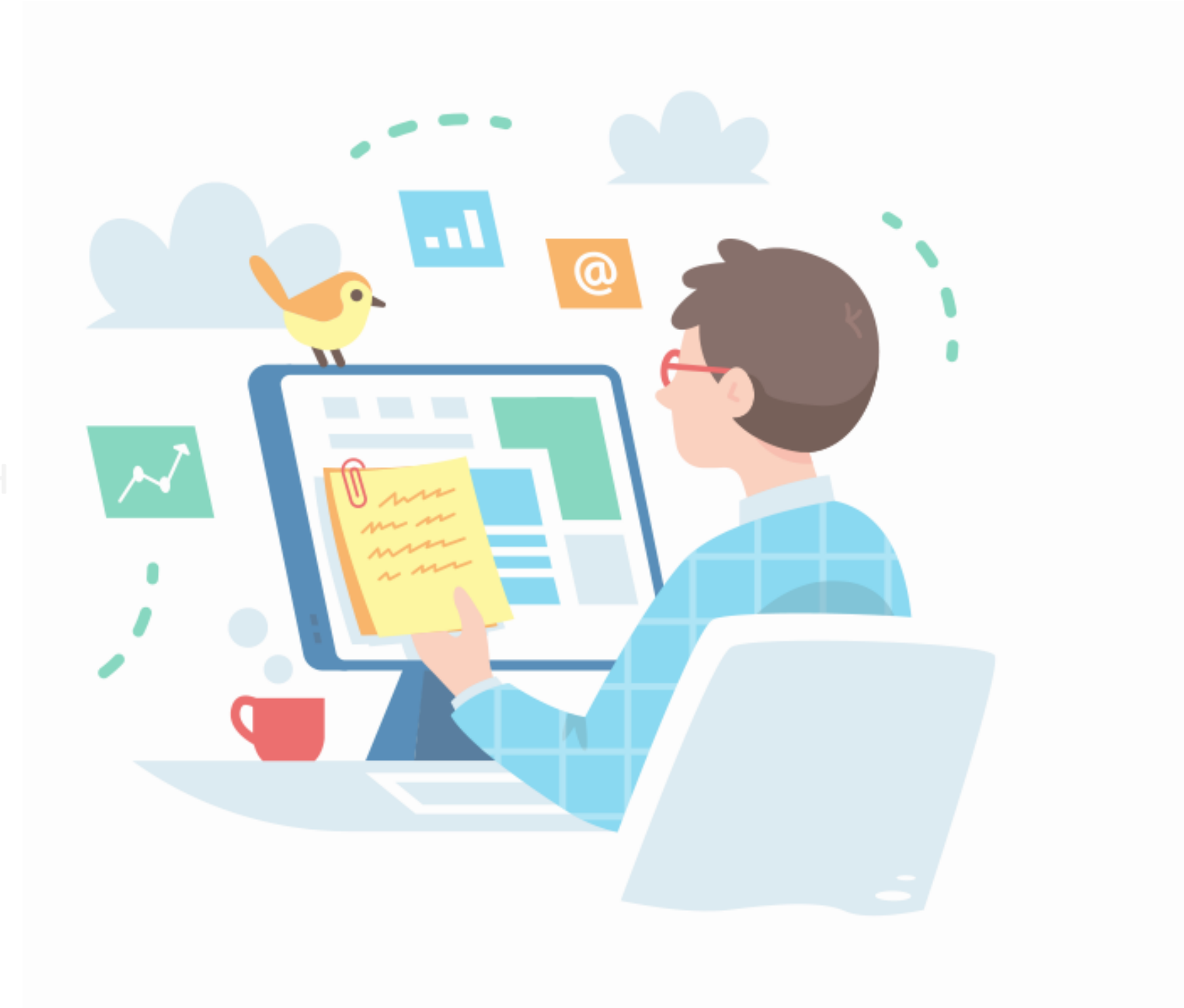
# BÀI TẬP NỘP : QUẢN LÝ NHÂN VIÊN - VERSION 1

Công ty ABC cần xây dựng ứng dụng quản lý thông tin và tính lương cho nhân viên. Thông tin mỗi nhân viên bao gồm: mã nhân viên, họ tên, ngày sinh, địa chỉ.

- Lương nhân viên: hệ số lương \* lương cơ bản

## Yêu cầu:

1. Khai báo các lớp cần thiết.
2. Cài đặt 5 constructor cho lớp nhân viên
3. Cài đặt tất cả getter và setter cho tất cả các thuộc tính. Lưu ý, thực hiện các bước kiểm tra dữ liệu, trước khi gán giá trị thuộc tính trong các setter.
4. Hãy cài đặt lớp CongTy để quản lý danh sách các nhân viên trong công ty, với các chức năng sau:
  - a. Nhập, xuất danh sách các nhân viên.
  - b. Tính tổng tiền lương của tất cả nhân viên.
  - c. Xuất nhân viên có lương cao nhất.





## NOW, Let's do them - Giới thiệu UML - [draw.io](https://draw.io) tool



Các bạn liệt kê LỚP đối tượng, các thuộc tính và phương thức có thể có cho các bài toán sau đây: (CODE TRÊN GIẤY hoặc [draw.io](https://draw.io) tool )

1. Mô tả lớp đối tượng cho phép thực hiện các phép tính Cộng, Trừ, Nhân, Chia các Phân Số
2. Mô tả lớp đối tượng cho phép quản lý Tài khoản ngân hàng của người dùng.
3. Mô tả lớp đối tượng cho phép quản lý thông tin Sinh Viên
4. Mô tả lớp đối tượng cho phép quản lý các Bệnh án điện tử
5. Mô tả lớp đối tượng cho phép quản lý thông tin Chuyến bay.
6. Mô tả lớp đối tượng cho phép quản lý thông tin Cầu thủ.
7. Mô tả lớp đối tượng cho phép quản lý thông tin Sổ tiết kiệm
8. Mô tả lớp đối tượng cho phép quản lý Lớp học ( thông tin lớp và DS Học sinh)
9. Mô tả lớp đối tượng cho phép quản lý Công ty (thông tin công ty và DS nhân viên)
10. Mô tả lớp đối tượng cho phép quản lý Trường học ( thông tin trường và DS các Lớp)



# Dự án 4

Ngân hàng ABC muốn lưu trữ thông tin của mỗi tài khoản như sau:

Mỗi tài khoản chứa các thông tin: Số tài khoản (Kiểu int), Tên tài khoản (kiểu chuỗi), Số tiền trong tài khoản (kiểu double)

(a). Thiết kế lớp Account để lưu trữ các thông tin, lớp bao gồm các phương thức sau:

- Constructor: Có 2 constructor (mặc định và đầy đủ tham số)
- Các phương thức get, set cho từng thuộc tính, nhập xuất.

(b). Thêm các thông tin sau vào lớp Account

- \* Hằng số lãi suất có giá trị khởi tạo 0.035
- \* Constructor có 2 đối số: số tài khoản, tên tài khoản. Constructor này sẽ khởi tạo số tiền mặc định là 50.
- Phương thức nạp tiền vào tài khoản: Lấy số tiền hiện tại trong tài khoản + số tiền nạp vào
- Phương thức rút tiền: Lấy số tiền hiện tại trong tài khoản – (số tiền muốn rút + phí rút tiền)
- Phương thức đáo hạn: Mỗi lần đến kỳ đáo hạn thì số tiền trong tài khoản = số tiền trong tài khoản + số tiền trong tài khoản \* LAISUAT
- Phương thức chuyển khoản từ tài khoản này sang tài khoản khác