**Bài Tập Quản Lý Nhân Viên Công Ty**  
Hãy xây dựng một chương trình quản lý danh bạ nhân viên trong công ty, yêu cầu sử dụng các kiến thức Java đã học từ Module 2 về lập trình hướng đối tượng, kế thừa, interface, xử lý ngoại lệ, và collections.

**Mô Tả Bài Tập**

1. **Tạo Class Employee:**
   * Tạo một lớp Employee với các thuộc tính:
     + name (String): Tên nhân viên.
     + age (int): Tuổi nhân viên.
     + salary (double): Lương của nhân viên.
   * Viết constructor để khởi tạo các thuộc tính trên.
   * Tạo phương thức displayInfo() để hiển thị thông tin của nhân viên, bao gồm tên, tuổi và lương.
2. **Tạo Class Manager kế thừa từ Employee:**
   * Lớp Manager sẽ kế thừa từ lớp Employee và có thêm thuộc tính:
     + department (String): Phòng ban mà quản lý này phụ trách.
   * Override phương thức displayInfo() trong lớp Manager để hiển thị thêm thông tin phòng ban (Department).
   * **Overloading**: Tạo thêm một phương thức displayInfo(String format) để cho phép hiển thị thông tin nhân viên với định dạng tùy chỉnh (ví dụ: in theo định dạng JSON, hoặc định dạng khác).
3. **Interface Workable:**
   * Tạo một interface **Workable** với phương thức work(), dùng để biểu thị công việc mà nhân viên hoặc quản lý cần làm.
   * Cả lớp Employee và Manager đều implement interface này và cài đặt phương thức work() để mô tả công việc của từng loại nhân viên (ví dụ: "Working as an employee" hoặc "Managing department").
   * Khi một nhân viên hoặc quản lý được tạo, phương thức work() sẽ được gọi để hiển thị công việc của họ.
4. **Functional Interface và Lambda Expressions:**
   * Tạo một functional interface SalaryCalculation với phương thức calculateSalary(Employee employee), dùng để tính toán lương cho nhân viên.
   * Sử dụng **lambda expressions** để tạo các đối tượng SalaryCalculation cho phép tính toán lương của nhân viên.
     + Ví dụ: Bạn có thể tạo một lambda để tính toán lương dựa trên mức lương cơ bản và phụ cấp.
5. **Exception Handling:**
   * Tạo phương thức updateSalary(Employee employee, double newSalary) trong lớp Employee:
     + Phương thức này sẽ nhận vào một đối tượng Employee và mức lương mới newSalary.
     + Nếu mức lương mới nhỏ hơn 0, phương thức sẽ ném một ngoại lệ IllegalArgumentException và thông báo lỗi “Salary cannot be negative”.
   * Sử dụng **try-catch** trong phương thức gọi để bắt và xử lý ngoại lệ. Hãy chắc chắn rằng chương trình không bị crash và tiếp tục chạy bình thường.
6. **Tạo 1 class Company trong đó có các yêu cầu sau:**
   * **List (ArrayList)**: Tạo một ArrayList để lưu trữ danh sách nhân viên trong công ty. Cung cấp các phương thức:
     + **Mặc định khởi tạo sẵn 3 nhân viên**
     + Thêm nhân viên mới vào danh sách.
     + Hiển thị tất cả nhân viên trong danh sách.
   * **Map (HashMap)**: Tạo một HashMap để lưu trữ tên nhân viên là **key** và mức lương của họ là **value**. Cung cấp các chức năng:
     + Thêm hoặc sửa đổi thông tin lương của nhân viên.
     + Tìm kiếm và hiển thị lương của một nhân viên theo tên.
   * **Set (HashSet)**: Tạo một HashSet để lưu trữ các phòng ban của công ty. Các phòng ban trong công ty phải là duy nhất và không trùng lặp.
     + Thêm phòng ban mới vào HashSet.
     + Hiển thị tất cả các phòng ban trong công ty.
   * Cung cấp chức năm sắp xếp danh sách nhân viên (ArrayList) theo lương từ cao xuống thấp.
   * Viết một phương thức tìm kiếm nhân viên theo tên trong danh sách và in ra thông tin của họ nếu tìm thấy.

Tạo class Main và xây dựng menu với các chức năng ở ý số 6