



Đề thi mẫu

Tạo 1 file Java có tên STT{Số thứ tự}_{Họ và tên}_{Mã đề}.java (VD: **STT1_LeVanAn_1.java**), trong đó có hàm main(). Các class khác (nếu có) để trong class này. Bài làm cho các câu đặt trong các hàm tương ứng: cau1(), cau2(), cau3(), cau4(), và cau5(). Hàm main() gọi các hàm này thông qua 1 đối tượng.

Câu 1 (3 điểm):

a. Tạo 2 giao diện (interface) IStaff có method work() và IStudent có method study(). Tạo lớp trừu tượng (abstract) People:

- Có thuộc tính chứa tên **name** (public) kiểu String
- Có thuộc tính tuổi **age** (private) kiểu int có hàm get set tương ứng.
- Có thuộc tính **ID** (protected) kiểu String
- Có thuộc tính **birthDate** (default) kiểu String hoặc LocalDate
- Có 1 phương thức trừu tượng void showInfo()
- Có hàm tạo với tham số là các thuộc tính nói trên.

b. Tạo lớp NEUStudent kế thừa (extend) lớp People và triển khai (implement) từ 2 giao diện IStaff và IStudent. Lớp này khai báo thêm 1 thuộc tính là studentID (protected String). Tạo 1 hàm tạo (constructor) cho class này với tham số là đầy đủ các thuộc tính. Chồng hàm (Override) showInfo() tại lớp này để hiển thị thông tin chi tiết sinh viên gồm toàn bộ các thuộc tính của sinh viên và thuộc tính của các lớp cha. Viết chồng hàm work() và study() cho lớp này.

c. Viết hàm **cau1()** tạo ra 3 đối tượng NEUStudent, bằng cách sử dụng hàm tạo đã viết. Trong đó sinh viên đầu tiên chính là mình (chính xác tên, mã sinh viên, ngày sinh, giới tính ..). Gọi hàm showInfo() của mỗi đối tượng để hiển thị thông tin của đối tượng.

Câu 2 (2 điểm):

Viết hàm **cau2()** thực hiện:

- a. Đọc vào số phần tử của mảng từ bàn phím, nhập các phần tử cho mảng và hỏi người dùng muốn hiện phần tử số mấy trong mảng rồi hiển thị phần tử đó ra màn hình.
- b. Lập trình đảm bảo bắt toàn bộ các ngoại lệ (exception) có thể xảy ra.

Câu 3 (2 điểm):

Viết hàm **cau3()** thực hiện

- a. Sử dụng class DataOutputStream để ghi 2 số nguyên vào 1 file có tên data.txt
- b. Sử dụng class DataInputStream để đọc 2 số nói trên và hiển thị tổng của 2 số đó ra màn hình

Câu 4 (2 điểm):

Viết hàm **cau4()** thực hiện:

- a. Tạo thread 1 để sau mỗi 5 giây hiển thị một số nguyên ngẫu nhiên
- b. Tạo thread 2 chạy song song liên tục yêu cầu người dùng nhập vào các con số và hiển thị tổng các số đã nhập. thread này dừng khi người dùng nhập vào số 0.

Câu 5 (1 điểm):

Viết hàm **cau5()** thực hiện:

Tạo ra một ArrayList, add 5 đối tượng bất kỳ vào ArrayList này. Thực hiện duyệt lặp để đọc các phần tử trong ArrayList và đưa vào 1 Map, key của Map là số tăng dần. Đọc Map này và hiển thị các phần tử của Map ra màn hình.

Sơ đồ class của bài làm

