

BÀI KIỂM TRA OOP 60 phút (đề 3)

Câu 1 (5đ). Sử dụng 1 JAVA IDE, viết lệnh khai báo các lớp kế thừa từ một lớp gốc (trừu tượng) và 1 interface mô tả đối tượng Person, TeachingAssistant, Lecturer, Professor như sau:

Lớp Person trừu tượng:

String name, int age, gender: public, có thể truy cập từ ngoài lớp.

String bank_account: protected, có thể truy cập từ các lớp con.

Hàm Constructor.

String getRole(): Phương thức trừu tượng.

Interface KPIEvaluator:

double calculateKPI(): Phương thức trong interface KPIEvaluator định nghĩa chức năng để tính KPI.

Lớp TeachingAssistant, Lecturer kế thừa (dẫn xuất) từ Person:

Lớp TeachingAssistant:

String employeeID: public (nếu đặt là private/protected thì cần hàm get/set).

private int numberOfCourses: lưu số lượng khóa học được hỗ trợ.

Constructor TeachingAssistant (String name, int age, String gender, String employeeID, int numberOfCourses): của lớp TeachingAssistant nhận thông tin đầu vào và gọi constructor của lớp cha Person để thiết lập thuộc tính kế thừa.

public String getRole(): Ghi đè phương thức getRole của lớp Person, để trả về chuỗi "**Teaching Assistant**".

public double calculateKPI(): Ghi đè phương thức calculateKPI() để tính KPI dựa trên số lượng khóa học hỗ trợ, mỗi khóa học tính 5 điểm.

Lớp Lecturer:

public String employeeID.

private int numberOfPublications: để lưu số lượng bài báo đã xuất bản.

Constructor Lecturer (String name, int age, String gender, String employeeID, int numberOfPublications): của lớp Lecturer nhận thông tin đầu vào và gọi constructor của lớp cha Person để thiết lập thuộc tính kế thừa.

public String getRole(): Ghi đè phương thức getRole của lớp Person, để trả về chuỗi "**Lecturer**".

public double calculateKPI(): Ghi đè phương thức calculateKPI() để tính KPI dựa trên số lượng bài báo. Mỗi bài báo đóng góp 7 điểm.

Lớp Professor kế thừa (dẫn xuất) từ Lecturer, và không cho phép có lớp kế thừa:

public static in countProfessors=0;

private int numberOfProjects: để lưu số lượng projects đã chủ trì.

Constructor Professor (String name, int age, String gender, String employeeID, int numberOfPublications, numberOfProjects): hàm tạo của lớp Professor nhận thông tin đầu vào và gọi constructor của lớp cha Lecturer để thiết lập thuộc tính kế thừa, tăng số **countProfessors** lên 1.

public String getRole(): Ghi đè phương thức getRole của lớp Person, để trả về chuỗi "**Professor**".

public double calculateKPI(): Ghi đè phương thức calculateKPI() để tính KPI dựa trên số lượng bài báo và số dự án đã chủ trì. Mỗi bài báo đóng góp 7 điểm, mỗi dự án đóng góp 10 điểm KPI.

Lớp chứa main():

Trong hàm main, tạo 1 hoặc 2 đối tượng **TeachingAssistant, Lecturer, Professor** và truyền thông tin vào.

+sử dụng phương thức **getRole()** và **calculateKPI()** để in thông tin vai trò và KPI của từng đối tượng.

Câu 2 (3đ). Lập trình nhập và hiển danh sách đối tượng.

2a.Viết hàm trong lớp chứa **main()**, nhập vào 1 danh sách các đối tượng:

Cách nhập mỗi đối tượng như sau:

+**nhập loại đối tượng chỉ 1 trong các xâu (không kể chữ hoa/thường) "ta","lec","gs" tương ứng đối tượng được tạo ra là TeachingAssistant, Lecturer, Professor**

+ **nhập các thông tin đầy đủ theo đối tượng. Nhập sai thì nhập lại.**

2b.Viết hàm trong lớp chứa **main()**, hiển thị danh sách đối tượng.

Gọi các hàm **2a, 2b** trên trong hàm main() để nhập và hiển thị danh sách đối tượng (thông tin, role, và KPI) đúng theo từng loại đối tượng **TeachingAssistant, Lecturer, Professor**.

2c. Hiển thị số Professor.

Câu 3 (2đ). Thiết kế UI (Chọn Controls, đặt Tab Order, chỉnh size và căn vị trí Controls):

3a. Thiết kế form gồm 1 label, 1 TextFiled ô nhập mã nhân sự (**mid**), 1 TextFiled ô nhập tuổi, 1 button Sửa và 1 button Thoát (chương trình được đóng khi nhấn nút **Thoát**), và 1 Table gồm 5 cột để hiển thị mảng đã nhập ở câu 2: Mã nhân sự, Họ tên, Tuổi, Giới tính, Chức danh.

Yêu cầu về UI (user interface) của form như sau:

+ **đặt tiêu đề form tiếng Việt.**

+ **đặt tab order phù hợp cho các điều khiển.**

+ **đặt vị trí và kích thước phù hợp cho các điều khiển.**

+ **đặt focus phù hợp khi form được load.**

+khi form được hiển thị thì **mid** được focus và rỗng, nút Sửa **không tương tác được**.

+lập trình đồ dữ liệu từ danh sách đối tượng nhập ở Console vào Table khi form được hiển thị.

3b.Lập trình nắm bắt các sự kiện:

+khi **mid** khác rỗng thì button Sửa sáng (tương tác được), trái lại mờ(không tương tác được).

+ **mid** chỉ cho phép nhập kí tự chữ số, nếu nhập sai không nhận.

Thao tác Sửa (có kiểm tra **valid** khi xử lý)

+ nhấn Sửa, khi **mid** có text độ dài là khác 4 thì hiển thị hộp thoại thông báo "Mã nhân sự không hợp lệ", đóng hộp thoại thì focus lại về **mid**.

+ nhấn Sửa, khi **mid** có text độ dài là 4, tìm dòng của Table có cột mã nhân sự trùng với nội dung ô **mid**, nếu không tìm thấy thì hiển thị hộp thoại thông báo "Mã nhân sự không tìm thấy", đóng hộp thoại thì focus lại về **mid**. Trái lại đặt dòng hiện tại của Table có nhân sự với mã được tìm thấy, sửa lại nội dung của cột tuổi theo nội dung ở TextField nhập tuổi và focus về **mid**.

Họ và tên sinh viên

Lớp,Mã sinh viên

Điểm bài làm.

Giảng viên

..... TS. Ngô Hoàng Huy

Ghi chú: Sinh viên được phép sử dụng tài liệu giấy để tham khảo các lớp/hàm chuẩn, chương trình mẫu.