

**ĐỀ KIỂM TRA GIỮA KỲ : LẬP TRÌNH JAVA – Mã đề: 01**

Thời gian: 80 phút

Xây dựng các lớp mô phỏng ứng dụng giao dịch ngân hàng theo mô tả như sau:

Class BankPolicy chứa trường tĩnh (static): interestRate lưu mức lãi suất ngân hàng theo năm

Class CustomerInfo có các trường: customerName, customerAdd và các phương thức khởi tạo hợp lý, inputData, setName, setAdd, getName, getAdd, toString, display

Class BankAccount kế thừa lớp CustomerInfo và có thêm các trường: accountNo, accountBalance, withdrawNum, depositNum, accountInterest và các phương thức khởi tạo hợp lý, inputData, doWithdraw, doDeposit, computeMonthInterest, transferMoney, toString, display

Class MainApp chứa trường accountList có kiểu: **Array**, **Vector**, **HashTable**, **ArrayList** và các phương thức sau: **initBank** cho phép tạo danh sách accountList từ: **bàn phím**, hoặc từ **tệp text** (mỗi hàng lưu thông tin một account: name \t add \t accountNo \t accountBalance); **display** cho phép hiển thị danh sách accountList; và các phương thức mô phỏng hoạt động giao dịch tại ngân hàng: **transferSimulation** cho phép chọn 2 tài khoản để thực hiện giao dịch chuyển tiền; **withdrawSimulation** cho phép chọn 1 tài khoản để thực hiện rút tiền; **depositSimulation** cho phép chọn 1 tài khoản để nộp tiền. VIết phương thức main trong lớp này và gọi các phương thức đã cài đặt của lớp.

[illegible]

## ĐỀ KIỂM TRA GIỮA KỲ : LẬP TRÌNH JAVA – Mã đề: 02

Thời gian: 80 phút

Xây dựng các lớp mô phỏng ứng dụng quản lý Sinh viên thực tập mô tả như sau:

Class trừu tượng Person chứa các trường: name, address khai báo protected hoặc public và các phương thức inputData, display, toString

Class `Lecturer` mô tả giảng viên là lớp kế thừa từ lớp `Person` và có thêm trường `lecId` kiểu `String` và các phương thức khởi tạo hợp lý, `inputData`, `setName`, `setAdd`, `getName`, `getAdd`, `toString`, `display`

Class Student mô tả sinh viên là lớp kế thừa từ lớp Person và có thêm các trường: `studentId`, `lecId`, `topicTitle`, `grade` và các phương thức khởi tạo hợp lý, phương thức `inputData`, `display`, `toString`, và các phương thức `setters` và `getters` cần thiết.

Class MainApp chứa các trường lecturerList và studentList kiểu **Array**, **ArrayList**, **Vector**, **Hashtable**, để lưu danh sách cán bộ và sinh viên; và các phương thức: initLecturers cho phép tạo danh lecturerList từ **bàn phím**, **từ tệp text** (mỗi hàng lưu thông tin một cán bộ, các trường cách nhau bởi dấu tab \t); phương thức initStudents cho phép tạo danh sách studentList từ **bàn phím**, **từ tệp text** t (mỗi hàng lưu thông tin một sinh viên, các trường cách nhau bởi dấu tab \t); phương thức displayLecturers, phương thức displayStudents; phương thức searchStudentByName; searchMaxGrade tìm sinh viên có điểm grade cao nhất. Viết phương thức main trong lớp này để tạo đối tượng của lớp MainApp và gọi các phương thức đã cài đặt của lớp.

[illegible]

**ĐỀ KIỂM TRA GIỮA KỲ : LẬP TRÌNH JAVA – Mã đề: 03**

Thời gian: 80 phút

Xây dựng các lớp mô phỏng quản lý danh sách cán bộ trường Đại học mô tả như sau:

Class University mô tả trường Đại học chứa các trường: uniName, uniAaddress khai báo protected hoặc public và các phương thức inputData, display, toString

Class Lecturer mô tả cán bộ có các trường: lecId, lecName, lecAddress và các phương thức khởi tạo hợp lý, inputData, setName, setAdd, getName, getAdd, toString, display

Class Faculty mô tả một khoa là lớp kế thừa từ lớp University và có thêm các trường: facultyId, facultyName, và lecturerList có kiểu **Array**, **ArrayList**, **Hashtable**, **Vector** để lưu danh sách cán bộ của khoa; và các phương thức: khởi tạo, phương thức initFaculty cho phép tạo danh sách cán bộ lecturerList từ **bàn phím**, **từ tệp text** (mỗi hàng lưu thông tin một lecturer, các trường cách nhau bởi dấu tab \t); phương thức display hiển thị danh sách cán bộ; phương thức searchByName tìm kiếm cán bộ theo tên. Viết phương thức main trong lớp này để tạo đối tượng của lớp Faculty và gọi các phương thức đã cài đặt của lớp.

[illegible]