



## Bài tập phần đọc ghi file

*Các lớp liên quan trong của từng bài tập sau cho vào cùng một gói. Bạn tự chia gói và đặt tên gói cho phù hợp. Tự thiết kế thêm các lớp, phương thức hỗ trợ(helper method, helper class) nếu cần. Đọc kĩ các ý của từng đề trước khi thực hiện.*

**Bài 1.** Thông tin về nhân viên bao gồm: mã nhân viên(cả chữ và số), họ tên đầy đủ, địa chỉ, tuổi, số điện thoại, mức lương, số năm kinh nghiệm. Hành động của nhân viên gồm làm việc, nghỉ ngơi, nhận lương, đi du lịch. Thiết kế lớp mô tả đối tượng nhân viên như trên. Trong đó thiết kế các phương thức khởi tạo cần thiết. Viết các cặp getter/setter cho từng thuộc tính. Cung cấp các đoạn chương trình để lọc và xét duyệt dữ liệu đầu vào trước khi gán giá trị cho các thuộc tính trong các setter nếu cần thiết.

Load toàn bộ dữ liệu hiện có từ file EMP.TXT ra danh sách khi chương trình được kích hoạt. Viết chương trình có menu cho phép chọn thực hiện các chức năng sau:

- Thêm mới một nhân viên vào danh sách sử dụng ArrayList. Lưu ý mã của mỗi nhân viên là duy nhất và không trùng nhau. Mã nhân viên đầu tiên bắt đầu với EMP1001.
- Hiển thị thông tin nhân viên ra màn hình theo dạng bảng gồm các hàng các cột.
- Tìm nhân viên theo tên.
- Xóa nhân viên theo mã cho trước.
- Ghi danh sách nhân viên vào file EMP.TXT, thông tin mỗi thuộc tính của từng đối tượng ghi trên 1 dòng. Lưu ý bảo toàn dữ liệu cũ của file(nếu có).
- Thoát chương trình.

**Bài 2.** Giả định thông tin về tài khoản ngân hàng ở Wakanda gồm: mã tài khoản, số tài khoản, tên tài khoản, loại tài khoản, số dư, tên ngân hàng phát hành, ngày phát hành, ngày hết hạn. Hành động có thể thực hiện gồm: nạp tiền vào tài khoản, rút tiền khỏi tài khoản, chuyển khoản, kiểm tra số dư, thanh toán các giao dịch khác như điện nước, vé máy bay, vé tàu du hành liên sao ... Thiết kế lớp với các phương thức phù hợp để mô tả các thông tin trên.

Load toàn bộ dữ liệu hiện có từ file ACC.DAT ra khi chương trình được kích hoạt. Lưu ý kiểm tra và xử lý các lỗi khi đọc ghi file. Viết chương trình có menu cho phép chọn thực hiện các chức năng sau:

- Thêm mới một tài khoản vào danh sách sử dụng ArrayList. Lưu ý một tài khoản chỉ có 1 mã duy nhất tự động tăng bắt đầu từ WAB100001.
- Hiển thị thông tin tài khoản ra màn hình theo dạng bảng gồm các hàng các cột.
- Nạp tiền vào tài khoản với mã TK cho trước. Lưu ý kiểm tra tính hợp lệ của số tiền cần nạp. Thông báo kết quả hành động này ra màn hình.
- Rút tiền khỏi tài khoản với mã TK cho trước. Lưu ý kiểm tra các vấn đề liên quan. Thông báo kết quả sau khi thực hiện việc rút tiền ra màn hình.
- Chuyển tiền từ TK A đến TK B theo số TK. Lưu ý kiểm tra các vấn đề liên quan. Thông báo kết quả sau khi chuyển tiền ra màn hình.



- f) Tìm tài khoản theo tên tài khoản.
- g) Tìm tài khoản theo mã tài khoản.
- h) Tìm tài khoản có số dư  $\geq x$  với  $x$  là giá trị nguyên nhập vào từ bàn phím.
- i) Xóa một tài khoản theo mã cho trước. Cho biết trạng thái sau khi xóa.
- j) Ghi danh sách tài khoản vào file ACC.DAT, thông tin mỗi tài khoản ghi trên 1 dòng. Mỗi thuộc tính phân tách nhau bằng 1 dấu -. Lưu ý bảo toàn dữ liệu cũ của file(nếu có).
- k) Thoát chương trình.

**Bài 3. (\*)** Thông tin môn học gồm: mã môn học, tên môn học, số tín chỉ, số tiết học, số bài kiểm tra. Thông tin sinh viên gồm: mã sinh viên, họ, đệm, tên, địa chỉ, email, giới tính, số điện thoại, lớp, khoa. Thông tin về lớp học gồm: mã lớp, tên lớp, phòng học, giờ học, môn học, sinh viên, bảng điểm. Thông tin về bảng điểm gồm: mã bảng điểm, điểm hệ số 1, hệ số 2, hệ số 3, điểm trung bình, học lực. Học lực là một enum gồm các giá trị: F(trượt môn), D(yếu), C(trung bình), B(khá), A(giỏi), A+(xuất sắc). Mức điểm như sau:

- ✓ Trượt nếu điểm TB  $< 4.0$
- ✓ Yếu nếu điểm TB từ 4.0 đến dưới 5.0
- ✓ Trung bình nếu điểm TB từ 5.0 đến dưới 6.5
- ✓ Khá nếu điểm TB từ 6.5 đến dưới 8.0
- ✓ Giỏi nếu điểm TB từ 8.0 đến dưới 9.0
- ✓ Xuất sắc nếu điểm TB từ 9.0 đến 10.0

Tự thiết kế lớp với các thuộc tính và phương thức phù hợp theo chuẩn tính đóng gói dữ liệu. Load toàn bộ dữ liệu trong các file dữ liệu SUB.DAT, STUD.DAT, COU.DAT, COU-TRAN.DAT, TRAN.DAT ra các danh sách tương ứng khi chương trình được kích hoạt. Viết chương trình có menu cho phép chọn thực hiện các chức năng sau:

- a) Thêm mới một môn học vào danh sách môn học sử dụng ArrayList. Mỗi môn học có mã duy nhất tự động tăng và bắt đầu từ mã SUB1001.
- b) Thêm mới một sinh viên vào danh sách sinh viên sử dụng ArrayList. Mỗi sinh viên có mã duy nhất tự động tăng và bắt đầu từ mã STU100001.
- c) Thêm mới một lớp học vào danh sách lớp học. Lưu ý kiểm tra mã môn học. Nếu mã môn không tồn tại thì không tạo mới lớp học. Mỗi lớp học có một mã duy nhất tự động tăng bắt đầu từ mã COU1001.
- d) Thêm sinh viên vào lớp học đã có bằng cách nhập mã lớp và nhập mã sinh viên cần thêm. Nếu mã lớp hoặc mã sinh viên không tồn tại thì không thêm được. Thông báo kết quả thực hiện ra màn hình. Mỗi sinh viên với mã SV X chỉ xuất hiện 1 lần trong 1 lớp học có mã lớp Y.
- e) Hiển thị danh sách môn học.
- f) Hiển thị danh sách sinh viên.
- g) Hiển thị danh sách lớp học.



- h) Nhập điểm và tính điểm trung bình cho từng sinh viên trong lớp. Công thức tính:  $TB = (1 \cdot \text{điểm hệ 1} + 2 \cdot \text{điểm hệ 2} + 3 \cdot \text{điểm hệ 3}) / 6$ ; Lưu vào bảng điểm. Mã của mỗi bảng điểm là duy nhất, tự động tăng bắt đầu từ mã TRA1001.
- i) Xét học lực cho từng sinh viên trong lớp.
- j) Hiển thị bảng điểm và học lực của tất cả các sinh viên trong 1 lớp có mã nhập vào từ bàn phím theo dạng bảng gồm các hàng và các cột. Danh sách được sắp xếp theo thứ tự giảm dần của điểm trung bình. Các thông tin hiển thị gồm: mã lớp, mã sinh viên, tên sinh viên, điểm hệ 1, 2, 3, TB và học lực. Thông tin mỗi sinh viên trên 1 dòng. Lưu ý kiểm tra kết quả và thông báo hợp lý nếu không có dữ liệu.
- k) Tìm xem sinh viên với mã cho trước có trong lớp học không.
- l) Ghi danh sách thông tin về môn học, sinh viên, lớp học, danh sách mã các sinh viên và mã bảng điểm tương ứng của mỗi lớp học, bảng điểm lần lượt ra các file tương ứng: SUB.DAT, STUD.DAT, COU.DAT, COU-TRAN.DAT, TRAN.DAT. Lưu ý bảo toàn dữ liệu cũ trong file(nếu có). Thông tin của mỗi đối tượng ghi trên 1 dòng. Các thuộc tính phân tách nhau bởi dấu \*.
- m) Thoát chương trình.

(\*) Bài khó dần để tăng level, skill(cho sát thực tế, như đầu khóa học đã nói: chúng tôi đào tạo lập trình viên, không đào tạo coder == thợ code).

Trang chủ: <https://braniumacademy.net/>