File giải thích thuật toán

Made by: Henry Bui

1. **J1.P0021**

* **Các thông tin của hàm student cần lưu ý**

+ Id: có thể lặp lại nhau do cùng một sinh viên có thể học nhiều khóa học hay học cùng khóa học trong khoảng thời gian khác nhau.

+ Student name: tên sinh viên

+ Semester: học kì

+ CourseName: tên khóa học, chỉ bao gồm C/C++, Java, .Net.

* **Hàm create student**

+ Cho người dùng nhập 10 sinh viên (vòng lặp for).

+ Mỗi lần lặp sẽ check từng thông tin của sinh viên thông qua class Checker và sử dụng regex, trường hợp duplicate tức cả 4 thông tin đều trùng lặp với một phần tử trong database, giảm i đi một đơn vị rồi yêu cầu người dùng nhập lại.

+ Check xem người dùng có muốn tiếp tục nhập thông tin hay không bằng cách chỉ cho phép nhập Y hoặc N, nếu có nhập lại, còn không, thoát khỏi hàm.

* **Hàm find sort student**

+ Kiểm tra xem database có trống không, sau đó tạo thêm một biến ArrayList tạm thời làm danh sách sinh viên đã được tìm thấy (**foundedList**)

+ Người dùng nhập vào bàn phím tên hoặc một từ khóa, sau đó dùng vòng lặp for truy cập từng phần tử để tìm kiếm thông tin của từng học sinh trong database xem studentName có chứa thông tin người dùng nhập vào không. Mỗi lần tìm thấy được một phần tử hợp lệ, sử dụng method add() để đưa biến Student vào foundedList.

+ nếu foundedList trống (không tìm thấy bất kì Student nào hợp lệ), thông báo người dùng rồi về màn hình chính.

+ Nếu foundedList còn, show các thông tin theo yêu cầu, không quan trọng hóa vấn đề interface do đề bài không yêu cầu.

* **Hàm update, delete**

+ Người dùng nhập vào bàn phím id hoặc một phần id

+ **Hàm con findStudent**: dùng vòng lặp for truy cập từng phần tử trong ArrayList của database tìm kiếm Student hợp lệ có id chứa thông tin người dùng cần tìm, sau đó, tạo thêm một arrayList để lưu các danh sách Student hợp lệ (foundedList), đưa các Student hợp lệ bằng method add(). In ra màn hình các thông tin của foundedList, người dùng nhập vào bàn phím một con số tương ứng với Student trên màn hình, xuất ra Student tương ứng.

+ có được student tương ứng, phần mềm sẽ hỏi người dùng có muốn update hoặc delete không, yêu cầu người dùng nhập U hoặc D tương ứng với update hoặc delete

+ nếu người dùng chọn Update, nhập các thông số mới (trừ id), check xem thông tin có bị lặp không, nếu có, thông báo với người dùng không có gì thay đổi rồi thoát, nếu không lưu lại thông tin rồi trở về màn hình chính.

* Hàm report:

+ Tạo thêm class report chứa các thông tin yêu cầu.

+ Dùng Collections sort() sắp xếp ArrayList theo mức độ ưu tiên: id -> CourseName.

+ Dùng vòng lặp for truy cập vào dãy đã được sắp xếp (sortedList) từ phần tử đầu tiên đến phần tử sortedList.size()-1, so sánh từng cặp liền kề nhau, nếu id và courseName trùng nhau, total tiến lên một đơn vị (total là giá trị int, mặc định bằng 1) cho đến khi khác nhau, đưa giá trị trùng nhau đó + total vào ArrayList có tên reportedList.

+ Đến cuối cùng để thêm giá trị cuối vào mảng reportedList, kiểm tra xem giá trị của total có bằng một không, nếu khác 1, tức là chưa gộp được các phần tử trước đó, tăng total lên thêm một đơn vị rồi đưa vào reportedList, nếu không (tức là phần tử cuối cùng khác so với phần tử trc đó), đưa vào reportedList với giá trị total mặc định là 1.

+ In ra màn hình danh sách reported theo yêu cầu của đề bài.

* Lưu ý: Mỗi lần show ra màn hình, tìm độ dài lớn nhất trong danh sách tìm được để display phù hợp với interface.
* Thứ tự test:
* Biến choice: nhấn enter ngay, kí tự, 6.7, -6, 0, 6, 5. (x)
* Create: (x)

+ id: enter, kí tự đặc biệt, kí tự hợp lệ.

+ studentName: enter, kí tự đặc biệt, kí tự hợp lệ

+ Semester: enter, kí tự đặc biệt, kí tự hợp lệ.

+ CourseName: enter, kí tự đặc biệt, một khóa học khác ngoài C/C++, Java,.Net, trường hợp hợp lệ.

+ Trường hợp duplicate: đưa vào trường hợp thông tin thứ 3 lặp lại thông tin thứ 2.

+ Nhập hợp lệ tiếp 7 thông tin còn lại.

+ Kiếm tra continue: enter, kí tự đặc biệt, kí tự hợp lệ.

* FindSortStudentByName: (x)

+ trường hợp database rỗng

+ keyword: enter, kí tự đặc biệt, kí tự hợp lệ

+ trường hợp không tìm thấy thông tin

* Update/delete student by id (x)

+ trường hợp database rỗng.

+ input: enter, kí tự đặc biệt, kí tự hợp lệ (không tồn tại, tồn tại).

+ nhập số: enter, hj,các giá trị biên, giá trị hợp lệ.

+ trường hợp sửa, trường hợp xóa.

* Report (x)

+ trường hợp database rỗng

+ trường hợp database đủ.

1. **Candidate Management J1.L.P0022**

* Các thông tin/chỉ số của class Candidate
* ID: check duplicate,…
* FirstName
* LastName
* BirthDate: năm sinh
* CandidateType
* Experience có:

+ ExpInYear: năm kinh nghiệm

+ ProSkill: chuyên ngành

* Fresher có:

+ Graduation\_date: ngày tốt nghiệp

+ Graduation\_rank: hạng tốt nghiệp

+ Education: trường đã tốt nghiệp.

* Intern có:

+ Majors

+ Semester

+ University Name