# - Factory:

Bài tập

Viết chương trình thực hiện kịch bản sau:

* đọc danh sách sinh viên từ *nguồn dữ liệu*
* insert một sinh viên vào *nguồn dữ liệu*
* xoá một sinh viên (dựa vào mã sinh viên) từ *nguồn dữ liệu*

Thông tin sinh viên bao gồm: mã sinh viên, họ tên, giới tính nam?, ngày sinh.

Nguồn dữ liệu có thể là **file văn bản**, hoặc là **một bảng trong cơ sở dữ liệu SQL Server**.

Mô tả thêm về hai loại nguồn dữ liệu này.

* 1. Nguồn dữ liệu file văn bản:

Nguồn dữ liệu lưu ở file văn bản "**sinhvien.txt**" có định dạng: mỗi sinh viên lưu trên một dòng, các trường thông tin cách nhau bởi dấu ','

mã sinh viên,họ tên,giới tính nam?,ngày sinh

Ví dụ:

T012001,Nguyễn Thị Hoa,False,2001/10/30

T012005,Trần Văn Huy,True,2000/02/29

T102002,Lê Văn Nam,True,2001/09/02

Việc thêm mới sinh viên được thực hiện bằng cách chèn thêm một dòng dữ liệu tương ứng với sinh viên theo định dạng như ở trên. Việc xoá sinh viên khỏi file văn bản dữ liệu được thực hiện bằng cách xoá dòng tương ứng khỏi file văn bản.

* 1. Nguồn dữ liệu là bảng trong cơ sở dữ liệu SQL Server

Nguồn dữ liệu lưu ở cơ sở dữ liệu là khá quen thuộc -- Đó là bảng SinhVien(#MaSinhVien, HoTen, GioiTinhNam, NgaySinh) trong đó cột MaSinhVien là khoá chính. Ngoài ra, kiểu dữ liệu của các cột là như sau:

* + MaSinhVien NVARCHAR(10), là khoá chính
  + HoTen NVARCHAR(50)
  + GioiTinhNam BIT
  + NgaySinh DATE

Việc thêm mới, xoá sinh viên vào bảng có thể được thực hiện bằng các câu truy vấn SQL tương ứng lên cơ sở dữ liệu.

* **Yêu cầu về nội dung**
  + Áp dụng mẫu Simple Factory để viết chương trình cho phép linh động thay đổi loại nguồn dữ liệu & chạy 3 bước trong kịch bản: đọc, insert, delete sinh viên (như mô tả ở đầu bài)
  + Áp dụng mẫu Singleton để bảo đảm chỉ dùng nhiều nhất một kết nối đến máy chủ cơ sở dữ liệu
  + Chương trình đọc file cấu hình có tên config.txt chứa thông tin cấu hình của chương trình. File này gồm 2 dòng

Dòng thứ nhất mô tả tên loại nguồn dữ liệu là TEXT\_FILE hoặc MS\_SQLSERVER

Dòng thứ hai chứa chuỗi, mô tả đường dẫn đến file nguồn dữ liệu (nếu loại nguồn dữ liệu là TEXT\_FILE), hoặc là connection string (nếu loại nguồn dữ liệu là MS\_SQLSERVER)

Dựa vào loại nguồn dữ liệu được mô tả ở file config.txt mà chương trình sẽ đọc dữ liệu ở nguồn theo kịch bản

* + Các class/interface đều phải nằm trong package với tên là \_mãSinhVien. Ví dụ sinh viên có mã sinh viên là T1029999 sẽ tạo package \_T1029999 để chứa các class SinhVien, ChuongTrinh, ... trong bài làm của mình.

# - Adapter

1. Viết thư viện SinhVienLib.JAR trong đó mô tả 2 class SinhVien và SinhVienReader SinhVien gồm các thuộc tính **mã sinh viên**, **họ đệm, tên**, **điểm trung bình** (thang điểm 10)

SinhVienReader có hàm public static List<SinhVien> docDanhSach(String filename); đọc danh sách sinh viên từ file văn bản có cấu trúc:

Mã sinh viên,Họ đệm,Tên,điểm trung bình

1. Viết thư viện CongNhanLib.JAR trong đó mô tả 2 class CongNhan và CongNhanReader CongNhan gồm các thuộc tính **họ tên**, **ngày bắt đầu vào làm việc**

CongNhanReader có hàm public static List<CongNhan> docDanhSach(String filename); đọc danh sách công nhân từ file văn bản có cấu trúc

họ tên,ngày bắt đầu vào làm việc

trong đó ngày bắt đầu vào làm việc có định dạng d/M/yyyy

1. Viết một chương trình thực hiện việc

* đọc danh sách sinh viên & công nhân ở 2 file dữ liệu sv.txt, cn.txt
* in ra màn hình danh sách các đối tượng sinh viên & công nhân, sắp xếp theo thứ tự ưu tiên dạng như ví dụ sau
* \*\*STT Họ tên Loại Độ ưu tiên\*\*
* 1 Nguyen Van SV A
* 2 Le Thi Hoa CN B
* 3 Le Van CN C
* 4 Tran Thi SV D
* 5 Le Thi CN x
* ...

trong đó, “**Độ ưu tiên**” được xác định là số X, D, C, B, A tương ứng

* + nếu là SinhVien —> Độ ưu tiên = được xếp theo thang điểm 4 (10 --> 4)
* < 4 → X
* 4 → 6.5 → D
* 6.5 → 7.5 → C
* 7.5 → 8.5 → B
* ≥ 8.5 → A
* nếu là Công nhân được xếp theo thâm niên làm việc
  + dưới 2 năm: X
  + từ 2 năm -> 5 năm: D
  + từ 5 năm -> 10 năm: C
  + từ 10 năm -> 20 năm: B
  + trên 20 năm: A

# - Composite:

Cần thiết kế cho tình huống với File, Folder.

* File có các thuộc tính tênFile (String)~~, ngàyChỉnhSửaCuối (DateTime)~~, kíchThước (Double).
* Folder gồm các thuộc tính tênThưMục (String)~~, ngàyChỉnhSửaCuối (DateTime)~~. Một Folder có thể chứa Folder con hoặc File với số lượng không hạn chế (0, 1, hoặc nhiều File, Folder).

**Yêu cầu**

a) Hãy tạo các lớp biểu diễn mô hình trên.

b) Hãy tạo đối tượng File, Folder theo cây phân cấp như sau:

c) Thử in ra các thông tin:

* Đường dẫn đầy đủ của file baitap02 → LAPTRINH/JAVA/baitap02
* Kích thước của file baitap02 → 3 kb
* Đường dẫn đầy đủ của thư mục LYTHUYET → LAPTRINH/JAVA/LYTHUYET
* Kích thước của thư mục LYTHUYET → 12 kb
* Kích thước của thư mục LAPTRINH → 21 kb

HOẶC:

Cho bảng `PhanTu(#Id, Ten, IdCha, DungLuong,Loai)`trong cơ sở dữ liệu  
  
dùng để lưu các `File`, `Folder` như bài học Mẫu thiết kế Composite.  
  
Hãy thiết kế chương trình thực hiện các công việc sau:  
  
1. Đọc toàn bộ cấu trúc cây thư mục được lưu trong cơ sở dữ liệu  
2. Nhập từ bàn phím một `Id`, sau đó in ra màn hình thông tin của phần tử tương ứng, bao gồm:  
1. Tên của đối tượng  
2. Đường dẫn đầy đủ của đối tượng (các thành phần của đường dẫn cách nhau bởi ký tự “/“)  
3. Loại đối tượng: là file hay folder  
4. Dung lượng của đối tượng  
3. Bổ sung thêm một đối tượng `File` tuỳ ý vào thư mục có `Id` nhập từ bàn phím. Sau khi bổ sung thành công thì đọc thông tin thư mục này từ cơ sở dữ liệu và hiển thị ra màn hình:  
1. Tên thư mục  
2. Đường dẫn đầy đủ của thư mục  
3. Danh sách các file (trực tiếp hoặc gián tiếp) của thư mục  
4. Dung lượng của thư mục  
  
Chú ý quy tắc làm bài, nộp bài:  
  
1. Cơ sở dữ liệu được đặt tên là `MaSinhVienDB`  
2. Nộp file ZIP chứa (1) file script của cơ sở dữ liệu, (2) mã nguồn của project, (3) hình minh hoạ chương trình được chạy cho mỗi câu trong đề bài. File ZIP đặt tên là `MaSinhVien-HọTênSinhViên.ZIP`

# - Decoration:

Quản lý thuê bao điện thoại. Thông tin về một gói thuê bao bao gồm:

* phí thuê bao (tháng): int
* số phút được gọi miễn phí: int
* số tin nhắn được gửi miễn phí: int
* mức phí gọi điện thoại/phút: int
* mức phí gửi tin nhắn: int

Có các loại thuê bao Basic, TOMATO, Sinh viên, COUPLE, VIP, ...

Gói khuyến mãi ABC (cho phép miễn phí nhắn tin trong tháng), XYZ (gọi nội mạng miễn phí), … Có thể apply nhiều gói khuyến mãi, nhiều lần (tuỳ theo quy định của nhà cung cấp dịch vụ) vào 1 thuê bao.

→ Gói khuyến mãi là decoration cho Thuê bao

# - Strategy:

1. **Bài tập củng cố kiến thức:** Hoàn thiện project benchmark 3 thuật toán Sắp xếp trong ghi chú bài giảng
2. **Bài nộp để cộng điểm:** Benchmark khả năng insert cùng một tập dữ liệu (ví dụ 1000 sinh viên ngẫu nhiên) cho 3 loại cơ sở dữ liệu khác nhau: MySQL, SQL Server, SQLite

Gợi ý:

* 1. Tạo class SinhVien gồm một số thuộc tính ví dụ: ID, HoTen, GioiTinh, NgaySinh, …
  2. Tham chiếu đến các thư viện driver dữ liệu MySQL, SQL Server, SQLite. Tạo sẵn CSDL, bảng SinhVien rỗng trên các CSDL này
  3. Tạo các class theo mẫu Strategy và thực hiện chèn 1000 record để so sánh

# - State:

Cho class Utils sau đây cài đặt các phương thức đa phương tiện (multimedia):

public class Utils {

public static void rung() {

// thực hiện rung thiết bị với mức độ cho trước

System.out.println(“Rung điện thoại”);

}

public static void phátÂmThanh(int âmLượng, int sốLần) {

// thực hiện phát âm thanh BEEP BEEP với âm lượng và

// số lần cho trước

System.out.println(“BEEP-BEEP”);

}

}

Hãy cài đặt lớp ĐiệnThoại, cho phép thiết lập thiết bị ở một trong các trạng thái Yên Lặng, Chuẩn, Ngoài Trời theo đó:

Sử dụng các lớp đã thiết kế, viết chương trình để minh hoạ