**Tên project:** 22T1020362\_SimpleAndFactoryMethod

# **a. Nội dung mã nguồn tất cả class, interface, file tài nguyên trong project**

Trong src/

## **Package: \_22T1020362**

### **Program.java: File chạy chương trình**

package \_22T1020362;

import java.io.IOException;

import java.sql.Date;

import java.time.LocalDate;

import java.util.Scanner;

import \_22T1020362.config.Config;

import \_22T1020362.dao.ISVDAO;

import \_22T1020362.factorymethod.\*;

import \_22T1020362.models.SinhVien;

import \_22T1020362.simplefactory.SVDAOFactory;

public class Program {

public static void main(String[] args) throws IOException {

Scanner sc = new Scanner(System.in);

//Simple Factory

Config cg = new Config();

String type = cg.getConfigType();

SVDAOFactory simpleDAO = new SVDAOFactory();

ISVDAO dao = simpleDAO.getDAO(type);

//Hoặc sử dụng Factory Method

ISVDAO dao2;

if(type.equals("MS\_SQLSERVER")) {

DBSVDAOFactory db = new DBSVDAOFactory();

dao2 = db.getDAO();

}

else if(type.equals("TEXT\_FILE")) {

TextFileSVDAOFactory txt = new TextFileSVDAOFactory();

dao2 = txt.getDAO();

}

else {

dao2 = null;

}

//Dùng simple factory

System.out.println("-----------------------------Đọc-----------------------------");

//read

System.out.println("Danh sách SV:");

System.out.println("[Mã sinh viên" + " - " + "Họ tên" + " - " + "Giới tính nam" + " - " + "Ngày sinh]");

for(SinhVien sv: dao.readSV()) {

System.out.println(sv.getMaSinhVien() + " - " + sv.getHoTen() + " - " + (sv.isGioiTinhNam() ? "True" : "False") + " - " + sv.getNgaySinh());

}

System.out.println("-----------------------------Thêm-----------------------------");

//insert

SinhVien sv = new SinhVien("22T1020485", "Lê Văn Hoàng", true, Date.valueOf(LocalDate.now()));

dao.insertSV(sv);

System.out.println("-----------------------------Xóa-----------------------------");

//delete

String msv = "22T1020485";

dao.deleteSV(msv);

sc.close();

//Dùng Factory Method

if(dao2 != null) {

System.out.println("-----------------------------Đọc Text File-----------------------------");

//read

System.out.println("Danh sách SV:");

System.out.println("[Mã sinh viên" + " - " + "Họ tên" + " - " + "Giới tính nam" + " - " + "Ngày sinh]");

for(SinhVien sv1: dao2.readSV()) {

System.out.println(sv1.getMaSinhVien() + " - " + sv1.getHoTen() + " - " + (sv1.isGioiTinhNam() ? "True" : "False") + " - " + sv1.getNgaySinh());

}

System.out.println("-----------------------------Thêm Text File-----------------------------");

//insert

SinhVien sv1 = new SinhVien("22T1020485", "Lê Văn Hoàng", true, Date.valueOf(LocalDate.now()));

dao2.insertSV(sv1);

System.out.println("-----------------------------Xóa Text File-----------------------------");

//delete

String msv1 = "22T1020485";

dao2.deleteSV(msv1);

}

}

}

## **Package: \_22T1020362.config**

### **Config.java: Đọc kiểu dữ liệu từ file “config.txt”**

package \_22T1020362.config;

import java.io.FileInputStream;

import java.io.IOException;

import java.util.Scanner;

public class Config {

public String getConfigType() throws IOException{

FileInputStream fis = new FileInputStream("config.txt");

Scanner sc = new Scanner(fis);

String line = sc.nextLine();

sc.close();

return line;

}

}

### **DBConfig.java: Áp dụng mẫu Singleton cho kết nối đến Hệ quản trị CSDL SQL Server**

package \_22T1020362.config;

import java.io.FileInputStream;

import java.sql.Connection;

import java.sql.DriverManager;

import java.util.Scanner;

public class DBConfig {

// 0. Khai báo các thuộc tính, hàm như bình thường

//Ket noi CSDL

private Connection cn;

public Connection getCn() {

return this.cn;

}

// 1. Khai báo private constructor

private DBConfig() {

try {

FileInputStream fis = new FileInputStream("config.txt");

Scanner sc = new Scanner(fis);

String line = null;

while(sc.hasNextLine()) {

line = sc.nextLine();

}

String [] lst = line.split("\\|");

cn = DriverManager.getConnection(lst[0], lst[1], lst[2]);

sc.close();

} catch (Exception e) {

e.printStackTrace();

}

}

// 2. Khai báo INSTANCE

private static DBConfig instance = null;

// 3. Khai báo getInstance

public static DBConfig getInstance() {

if (instance == null) {

instance = new DBConfig();

}

return instance;

}

}

### **TextFileConfig.java: Áp dụng mẫu Singleton cho kết nối đến file text**

package \_22T1020362.config;

import java.io.FileInputStream;

import java.util.Scanner;

public class TextFileConfig {

private String txtfileUrl;

public String getTextFileUrl() {

return this.txtfileUrl;

}

private TextFileConfig() {

try {

FileInputStream fis = new FileInputStream("config.txt");

Scanner sc = new Scanner(fis);

String line = null;

while(sc.hasNextLine()) {

line = sc.nextLine();

}

txtfileUrl = line;

sc.close();

// mất 10s!!!

} catch (Exception e) {

e.printStackTrace();

}

}

private static TextFileConfig instance = null;

public static TextFileConfig getInstance() {

if (instance == null) {

instance = new TextFileConfig();

}

return instance;

}

}

## **Package: \_22T1020362.dao**

### **DBSVDAO.java: chứa các phương thức để truy vấn CSDL trong hệ quản trị CSDL SQL Server, bao gồm đọc, thêm, xóa sinh viên**

package \_22T1020362.dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

import \_22T1020362.models.SinhVien;

import lombok.\*;

@AllArgsConstructor

public class DBSVDAO implements ISVDAO{

private Connection cn;

@Override

public List<SinhVien> readSV(){

List<SinhVien> lst = new ArrayList<>();

String sql = "SELECT \* FROM SinhVien";

Connection cn = this.cn;

try {

PreparedStatement pr = cn.prepareStatement(sql);

ResultSet rs = pr.executeQuery();

while(rs.next()) {

SinhVien s = new SinhVien(rs.getString(1), rs.getString(2), rs.getBoolean(3), rs.getDate(4));

lst.add(s);

}

rs.close(); pr.close();

}

catch(Exception e) {

e.printStackTrace();

}

return lst;

}

@Override

public void insertSV(SinhVien sv){

String sql1 = "SELECT \* FROM SinhVien WHERE MaSinhVien LIKE " + "'" + sv.getMaSinhVien() + "'";

String sql2 = "INSERT INTO SinhVien (MaSinhVien, HoTen, GioiTinhNam, NgaySinh) VALUES (?, ?, ?, ?)";

Connection cn = this.cn;

try {

PreparedStatement pr = cn.prepareStatement(sql1);

ResultSet rs = pr.executeQuery();

if(rs.isBeforeFirst()) {

System.out.println("Mã sinh viên đã tồn tại trong hệ thống, quý khách xin vui lòng kiểm tra lại.");

}

else {

pr = cn.prepareStatement(sql2);

pr.setString(1, sv.getMaSinhVien());

pr.setString(2, sv.getHoTen());

pr.setBoolean(3, sv.isGioiTinhNam());

pr.setDate(4, sv.getNgaySinh());

int rowInserted = pr.executeUpdate();

if(rowInserted > 0) {

System.out.println("Thêm mới sinh viên có mã: " + sv.getMaSinhVien() + " thành công");

}

else {

System.out.println("Lỗi không xác định, hệ thống đang kiểm tra lại, xin quý khách vui lòng đợi trong giây lát");

}

}

rs.close();

pr.close();

}

catch(Exception e) {

e.printStackTrace();

}

}

@Override

public void deleteSV(String msv){

String sql1 = "SELECT \* FROM SinhVien WHERE MaSinhVien LIKE " + "'" + msv + "'";

String sql2 = "DELETE FROM SinhVien WHERE MaSinhVien LIKE ?";

Connection cn = this.cn;

try {

PreparedStatement pr = cn.prepareStatement(sql1);

ResultSet rs = pr.executeQuery();

if(!rs.isBeforeFirst()) {

System.out.println("Mã sinh viên không tồn tại trong hệ thống, quý khách xin vui lòng kiểm tra lại.");

}

else {

pr = cn.prepareStatement(sql2);

pr.setString(1, msv);

int rowInserted = pr.executeUpdate();

if(rowInserted > 0) {

System.out.println("Xóa thành công sinh viên có mã: " + msv);

}

else {

System.out.println("Lỗi không xác định, hệ thống đang kiểm tra lại, xin quý khách vui lòng đợi trong giây lát");

}

}

rs.close();

pr.close();

}

catch(Exception e) {

e.printStackTrace();

}

}

}

### **ISVDAO.java: Interface cho các DAO của DB và Text**

package \_22T1020362.dao;

import java.util.List;

import \_22T1020362.models.SinhVien;

public interface ISVDAO {

List<SinhVien> readSV();

void insertSV(SinhVien sv);

void deleteSV(String msv);

}

### **TextFileSVDAO.java: chức các phương thức để truy vấn dữ liệu trong file text, đọc thêm xóa sinh viến**

package \_22T1020362.dao;

import java.io.BufferedWriter;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileWriter;

import java.io.IOException;

import java.sql.Date;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

import \_22T1020362.models.SinhVien;

import lombok.\*;

@AllArgsConstructor

public class TextFileSVDAO implements ISVDAO{

private String fileUrl;

@Override

public List<SinhVien> readSV() {

List<SinhVien> list = new ArrayList<>();

FileInputStream fis = null;

try {

fis = new FileInputStream(fileUrl);

} catch (FileNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

Scanner sc = new Scanner(fis);

String line;

while(sc.hasNextLine()) {

line = sc.nextLine();

if(line.isEmpty()) break;

String [] lst = line.split(",");

SinhVien sv = new SinhVien(lst[0], lst[1] , Boolean.parseBoolean(lst[2]) , Date.valueOf(lst[3]));

list.add(sv);

}

sc.close();

return list;

}

@Override

public void insertSV(SinhVien sv) {

List<SinhVien> lst = readSV();

boolean check = false;

for(SinhVien s: lst) {

if(s.getMaSinhVien().equals(sv.getMaSinhVien())) {

System.out.println("Mã sinh viên đã tồn tại trong hệ thống, quý khách xin vui lòng kiểm tra lại.");

check = true;

}

}

if(!check) {

try {

FileWriter fw;

fw = new FileWriter(fileUrl, true);

BufferedWriter bw = new BufferedWriter(fw);

bw.write(sv.toString());

bw.newLine();

bw.close();

System.out.println("Thêm mới sinh viên có mã: " + sv.getMaSinhVien() + " thành công");

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

@Override

public void deleteSV(String msv) {

List<SinhVien> lst = readSV();

boolean removed = lst.removeIf(sv -> sv.getMaSinhVien().equals(msv));

if(removed) {

FileWriter fw;

try {

fw = new FileWriter(fileUrl, false);

BufferedWriter bw = new BufferedWriter(fw);

for(SinhVien sv: lst) {

bw.write(sv.toString());

bw.newLine();

}

bw.close();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

System.out.println("Hệ thống đã xóa sinh viên có mã: " + msv);

}

else {

System.out.println("Hệ thống không tìm thấy sinh viên có mã: " + msv);

}

}

}

## **Package: \_22T1020362.factorymethod**

### **DBSVDAOFactory.java: Factory cho SinhVien DAO - DB, áp dụng mẫu Factory Method**

package \_22T1020362.factorymethod;

import \_22T1020362.config.DBConfig;

import \_22T1020362.dao.DBSVDAO;

import \_22T1020362.dao.ISVDAO;

public class DBSVDAOFactory extends MethodSVDAOFactory {

@Override

public ISVDAO getDAO() {

return new DBSVDAO(DBConfig.getInstance().getCn());

}

}

### **MethodSVDAOFactory.java: Abstract Factory cho 2 loại Factory DB và Text, áp dụng mẫu Factory Method**

package \_22T1020362.factorymethod;

import \_22T1020362.dao.ISVDAO;

public abstract class MethodSVDAOFactory {

public abstract ISVDAO getDAO();

}

### **TextFileSVDAOFactory.java: Factory cho SinhVien DAO – Text, áp dụng mẫu Factory Method**

package \_22T1020362.factorymethod;

import \_22T1020362.config.TextFileConfig;

import \_22T1020362.dao.ISVDAO;

import \_22T1020362.dao.TextFileSVDAO;

public class TextFileSVDAOFactory extends MethodSVDAOFactory {

@Override

public ISVDAO getDAO() {

return new TextFileSVDAO(TextFileConfig.getInstance().getTextFileUrl());

}

}

## **Package: \_22T1020362.models**

### **SinhVien.java: chứa class SinhVien, các thuộc tính, hàm khởi tạo, hàm getter, setter, hàm riêng áp dụng Lombok để code thuận tiện**

package \_22T1020362.models;

import java.sql.Date;

import lombok.\*;

@Data

@AllArgsConstructor

@NoArgsConstructor

public class SinhVien {

private String maSinhVien;

private String hoTen;

private boolean gioiTinhNam;

private Date ngaySinh;

public String toString() {

return maSinhVien + "," + hoTen + "," + (gioiTinhNam ? "True" : "False") + "," + ngaySinh;

}

}

## **Package: \_22T1020362.simplefactory**

### **SVDAOFactory.java: Simple Factory cho SinhVien DAO của DB và Text, áp dụng mẫu Simple Factory, chia case trong đó**

package \_22T1020362.simplefactory;

import \_22T1020362.config.DBConfig;

import \_22T1020362.config.TextFileConfig;

import \_22T1020362.dao.DBSVDAO;

import \_22T1020362.dao.ISVDAO;

import \_22T1020362.dao.TextFileSVDAO;

public class SVDAOFactory{

public ISVDAO getDAO(String type) {

//thay đổi loại nguồn dữ liệu (Factory)

//& chạy 3 bước đọc thêm xóa trong kịch bản (ISVDAO, DBSVDAO, FileSVDAO)

switch (type) {

case "MS\_SQLSERVER":

return new DBSVDAO(DBConfig.getInstance().getCn());

case "TEXT\_FILE":

return new TextFileSVDAO(TextFileConfig.getInstance().getTextFileUrl());

default:

throw new IllegalArgumentException("Không hỗ trợ nguồn dữ liệu: " + type);

}

}

}

## **pom.xml:**

<project xmlns="https://maven.apache.org/POM/4.0.0" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>22T1020362</groupId>

<artifactId>22T1020362\_SimpleAndFactoryMethod</artifactId>

<version>0.0.1-SNAPSHOT</version>

<build>

<sourceDirectory>src</sourceDirectory>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.13.0</version>

<configuration>

<release>22</release>

</configuration>

</plugin>

</plugins>

</build>

<dependencies>

<dependency>

<groupId>com.microsoft.sqlserver</groupId>

<artifactId>mssql-jdbc</artifactId>

<version>12.8.1.jre11</version>

</dependency>

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<version>1.18.30</version>

<scope>provided</scope>

</dependency>

</dependencies>

</project>

# **b. Hình minh họa chạy chương trình (Cửa sổ Eclipse)**

A screenshot of a computer

Description automatically generated

# **c. Script của cơ sở dữ liệu -- bao gồm cả lược đồ & dữ liệu**

Trong database/

## **script.sql:**

USE [master]

GO

/\*\*\*\*\*\* Object: Database [22T1020362\_SimpleAndMethodFactory] Script Date: 21/03/2025 8:50:26 CH \*\*\*\*\*\*/

CREATE DATABASE [22T1020362\_SimpleAndMethodFactory]

CONTAINMENT = NONE

ON PRIMARY

( NAME = N'22T1020362', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\22T1020362.mdf' , SIZE = 4096KB , MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB )

LOG ON

( NAME = N'22T1020362\_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\22T1020362\_log.ldf' , SIZE = 1024KB , MAXSIZE = 2048GB , FILEGROWTH = 10%)

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET COMPATIBILITY\_LEVEL = 120

GO

IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))

begin

EXEC [22T1020362\_SimpleAndMethodFactory].[dbo].[sp\_fulltext\_database] @action = 'enable'

end

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET ANSI\_NULL\_DEFAULT OFF

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET ANSI\_NULLS OFF

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET ANSI\_PADDING OFF

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET ANSI\_WARNINGS OFF

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET ARITHABORT OFF

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET AUTO\_CLOSE OFF

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET AUTO\_SHRINK OFF

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET AUTO\_UPDATE\_STATISTICS ON

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET CURSOR\_CLOSE\_ON\_COMMIT OFF

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET CURSOR\_DEFAULT GLOBAL

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET CONCAT\_NULL\_YIELDS\_NULL OFF

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET NUMERIC\_ROUNDABORT OFF

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET QUOTED\_IDENTIFIER OFF

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET RECURSIVE\_TRIGGERS OFF

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET DISABLE\_BROKER

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET AUTO\_UPDATE\_STATISTICS\_ASYNC OFF

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET DATE\_CORRELATION\_OPTIMIZATION OFF

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET TRUSTWORTHY OFF

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET ALLOW\_SNAPSHOT\_ISOLATION OFF

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET PARAMETERIZATION SIMPLE

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET READ\_COMMITTED\_SNAPSHOT OFF

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET HONOR\_BROKER\_PRIORITY OFF

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET RECOVERY FULL

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET MULTI\_USER

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET PAGE\_VERIFY CHECKSUM

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET DB\_CHAINING OFF

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET FILESTREAM( NON\_TRANSACTED\_ACCESS = OFF )

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET TARGET\_RECOVERY\_TIME = 0 SECONDS

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET DELAYED\_DURABILITY = DISABLED

GO

EXEC sys.sp\_db\_vardecimal\_storage\_format N'22T1020362\_SimpleAndMethodFactory', N'ON'

GO

USE [22T1020362\_SimpleAndMethodFactory]

GO

/\*\*\*\*\*\* Object: Table [dbo].[SinhVien] Script Date: 21/03/2025 8:50:27 CH \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[SinhVien](

[MaSinhVien] [nvarchar](10) NOT NULL,

[HoTen] [nvarchar](50) NOT NULL,

[GioiTinhNam] [bit] NOT NULL,

[NgaySinh] [date] NOT NULL,

**CONSTRAINT** [PK\_SinhVien] **PRIMARY KEY** CLUSTERED

(

[MaSinhVien] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

USE [master]

GO

ALTER DATABASE [22T1020362\_SimpleAndMethodFactory] SET READ\_WRITE

GO

# **d. File minh hoạ nguồn dữ liệu TEXT**

## **config.txt**

MS\_SQLSERVER

jdbc:sqlserver://localhost:1433;databaseName=22T1020362\_SimpleAndMethodFactory;trustServerCertificate=true;encrypt=true|sa|sa

## **sinhvien.txt**

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

T012005,Trần Văn Huy,True,2000-02-29

T102002,Lê Văn Nam,True,2001-09-02