1. **Nội dung class SinhVien**

**package** T1020625;

**import** lombok.AllArgsConstructor;

**import** lombok.Getter;

**import** lombok.NoArgsConstructor;

@AllArgsConstructor

@NoArgsConstructor

@Getter

**public** **class** SinhVien {

**private** String maSinhVien;

**private** String hoTen;

**boolean** gioiTinhNam;

String ngaySinh;

}

1. **Nội dung interface SinhVienDAO**

**package** T1020625;

**import** java.io.IOException;

**import** java.text.ParseException;

**import** java.util.ArrayList;

**public** **interface** SinhVienDAO {

ArrayList<SinhVien> docDanhSachSinhVien() **throws** IOException, ParseException;

**public** **boolean** insertSinhVien (SinhVien sv) ;

**public** **boolean** deleteSinhVien (String maSinhVien) **throws** IOException;

}

1. **Nội dung class SinhVienFactory**

**package** T1020625;

**public** **class** SinhVienFactory {

**public** SinhVienDAO getPhuongThuc (String type) {

**if** (type=="txt") {

**return** **new** txtSinhVienDAO();

}

**else** {

**return** **new** DBSinhVienDAO();

}

}

}

1. **Nội dung class txtSinhVienDAO**
   1. **Phần khai báo và hàm readDanhSachSinhVien**

**package** T1020625;

**import** java.io.BufferedWriter;

**import** java.io.FileWriter;

**import** java.io.IOException;

**import** java.util.ArrayList;

**public** **class** txtSinhVienDAO **implements** SinhVienDAO {

String source = "src/sinhvien.txt";

@Override

**public** ArrayList<SinhVien> readDanhSachSinhVien() **throws** IOException {

ArrayList<SinhVien> sv = **new** ArrayList<>();

ArrayList<String> lines = FileHandler.*readAll*(source);

**for** (String line:lines) {

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

**if** (part.length == 4) {

SinhVien x = **new** SinhVien(part[0], part[1],

Boolean.*parseBoolean*(part[2]), part[3]);

sv.add(x);}

}

**return** sv;

}

* 1. **Hàm insertSinhVien**
  2. **Hàm isExisted**

**public** **boolean** insertSinhVien (SinhVien sv) {

**if** (isExisted (sv.getMaSinhVien())) **return** **false**;

**try** (BufferedWriter writer = **new** BufferedWriter(**new** FileWriter(source, **true**))) {

writer.write("\n" + sv.getMaSinhVien() + "," + sv.getHoTen() + "," + sv.isGioiTinhNam() + "," + sv.getNgaySinh());

**return** **true**;

} **catch** (IOException e) {

**return** **false**;

}

}

Boolean isExisted (String maSV) {

ArrayList<String> lines;

**try** {

lines = FileHandler.*readAll*("src/sinhvien.txt");

} **catch** (IOException e) {

**return** **false**;

}

**for** (String line : lines) {

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

**if** (part.length==4) {

**if** (part[0].equals(maSV)) **return** **true**;

}

}

**return** **false**;

}

* 1. **Hàm deleteSinhVien**

@Override

**public** **boolean** deleteSinhVien(String maSinhVien) **throws** IOException {

ArrayList<String> sv = **new** ArrayList<>();

**boolean** isDeleted = **false**;

ArrayList <String> lines = FileHandler.*readAll*(source);

**for** (String line: lines) {

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

**if** (part.length ==4) {

**if** (part[0].equals(maSinhVien)) isDeleted = **true**;

**else** sv.add(line);

}

}

**if** (isDeleted) {

FileHandler.*addAll*(sv, source); }

**return** isDeleted;

}

1. **Nội dung class FileHandler**

**package** T1020625;

**import** java.io.BufferedReader;

**import** java.io.BufferedWriter;

**import** java.io.FileReader;

**import** java.io.FileWriter;

**import** java.io.IOException;

**import** java.util.ArrayList;

**public** **class** FileHandler {

**public** **static** ArrayList <String> readAll(String src) **throws** IOException {

ArrayList <String> lines = **new** ArrayList <>();

BufferedReader br = **new** BufferedReader (**new** FileReader(src));

String line;

**while** ((line=br.readLine())!=**null**) {

lines.add(line);

}

br.close();

**return** lines;

}

**public** **static** **void** addAll(ArrayList<String> arr, String path) **throws** IOException {

BufferedWriter bw = **new** BufferedWriter (**new** FileWriter (path));

**for** (String line : arr) {

bw.write(line);

bw.newLine();

}

bw.close();

}

}

1. **Nội dung hàm DBSinhVienDAO**
   1. **Phần khai báo và hàm readDanhSachSinhVien**

**package** T1020625;

**import** java.io.IOException;

**import** java.sql.\*;

**import** java.text.SimpleDateFormat;

**import** java.util.ArrayList;

**public** **class** DBSinhVienDAO **implements** SinhVienDAO {

@Override

**public** ArrayList<SinhVien> readDanhSachSinhVien() **throws** IOException {

ArrayList<SinhVien> lstSV = **new** ArrayList<>();

Connection cnn = **null**;

Statement stmt = **null**;

ResultSet rs = **null**;

String sql ="select \* from SinhVien";

**try** {

**try** {

cnn = KetNoiGetter.*getInstance*().getCnn();

} **catch** (ClassNotFoundException e) {e.printStackTrace();}

stmt = cnn.createStatement();

rs = stmt.executeQuery(sql);

**while** (rs.next()) {

String maSinhVien = rs.getString("MaSinhVien");

String hoTen = rs.getString("HoTen");

**boolean** gioiTinhNam = rs.getBoolean("GioiTinhNam");

String ngaySinh = (rs.getDate("NgaySinh")).toString();

SinhVien sinhVien = **new** SinhVien(maSinhVien, hoTen, gioiTinhNam, ngaySinh);

lstSV.add(sinhVien);

}

} **catch** (SQLException e) {e.printStackTrace(); }

**finally** {

**try** {

**if** (rs != **null**) rs.close();

**if** (stmt != **null**) stmt.close();

**if** (cnn != **null**) cnn.close();

} **catch** (SQLException e) { e.printStackTrace();}

}

**return** lstSV;

}

* 1. **Hàm insertSinhVien**

@Override

**public** **boolean** insertSinhVien(SinhVien sv) {

Connection cnn = **null**;

PreparedStatement pstmt = **null**;

**try** {

cnn = KetNoiGetter.*getInstance*().getCnn();

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

pstmt = cnn.prepareStatement(sql);

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

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

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

SimpleDateFormat sdf = **new** SimpleDateFormat("yyyy-MM-dd");

java.util.Date utilDate = sdf.parse(sv.getNgaySinh());

java.sql.Date sqlDate = **new** java.sql.Date(utilDate.getTime());

pstmt.setDate(4, sqlDate);

**int** rowsInserted = pstmt.executeUpdate();

**return** rowsInserted >=1;

} **catch** (Exception e) {

**return** **false**;

} **finally** {

**try** {

**if** (pstmt != **null**) pstmt.close();

**if** (cnn != **null**) cnn.close();

} **catch** (SQLException ex) {

**return** **false**;

}

}

}

* 1. **Hàm deleteSinhVien**

**public** **boolean** deleteSinhVien(String maSinhVien) {

Connection cnn = **null**;

PreparedStatement pstmt = **null**;

**try** {

cnn = KetNoiGetter.*getInstance*().getCnn();

String sql = "DELETE FROM SinhVien WHERE MaSinhVien = ?";

pstmt = cnn.prepareStatement(sql);

pstmt.setString(1,maSinhVien);

**int** rowsInserted = pstmt.executeUpdate();

**return** rowsInserted >=1;

} **catch** (Exception e) {

**return** **false**;

} **finally** {

**try** {

**if** (pstmt != **null**) pstmt.close();

**if** (cnn != **null**) cnn.close();

} **catch** (SQLException ex) {

**return** **false**;

}

}

}

}

1. **Nội dung class KetNoiGetter**

**package** T1020625;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

**public** **class** KetNoiGetter {

**private** Connection cnn = **null**;

**public** Connection getCnn(){

**return** cnn;

}

**private** **static** KetNoiGetter *instance* = **null**;

**private** KetNoiGetter() **throws** SQLException, ClassNotFoundException {

Class.*forName*("com.microsoft.sqlserver.jdbc.SQLServerDriver");

String url = "jdbc:sqlserver://localhost:1433;databaseName=ViDuDB1;user=sa;password=sa;trustServerCertificate=true;";

cnn = DriverManager.*getConnection*(url);

}

**public** **static** KetNoiGetter getInstance() **throws** SQLException, ClassNotFoundException {

**if** (*instance* == **null** || *instance*.cnn.isClosed()) {

*instance* = **new** KetNoiGetter();

}

**return** *instance*;

}

}

**package** T1020625;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

**public** **class** KetNoiGetter {

**private** Connection cnn = **null**;

**public** Connection getCnn(){

**return** cnn;

}

**private** **static** KetNoiGetter *instance* = **null**;

**private** KetNoiGetter() **throws** SQLException, ClassNotFoundException {

Class.*forName*("com.microsoft.sqlserver.jdbc.SQLServerDriver");

String url = "jdbc:sqlserver://localhost:1433;databaseName=ViDuDB1;user=sa;password=sa;trustServerCertificate=true;";

cnn = DriverManager.*getConnection*(url);

}

**public** **static** KetNoiGetter getInstance() **throws** SQLException, ClassNotFoundException {

**if** (*instance* == **null** || *instance*.cnn.isClosed()) {

*instance* = **new** KetNoiGetter();

}

**return** *instance*;

}

}

1. **Nội dung class ChuongTrinh**
   1. **khai báo và hàm hienThiDanhSach**

**package** T1020625;

**import** java.io.IOException;

**import** java.util.ArrayList;

**public** **class** ChuongTrinh {

**private** **static** **void** hienThiDanhSach(ArrayList<SinhVien> danhSach) {

**for** (SinhVien sv : danhSach) {

System.***out***.println(

sv.getMaSinhVien() + " " +

sv.getHoTen() + " " +

sv.isGioiTinhNam() + " " +

sv.getNgaySinh()

);

}

}

* 1. **các hàm thêm và xoá sinh viên từ file txt và từ database**

**private** **static** **void** themSinhVienTXT(SinhVienDAO daotxt, SinhVien sv) {

**boolean** result = daotxt.insertSinhVien(sv);

System.***out***.println(result ? "Đã thêm vào TXT" : "Lỗi thêm vào TXT");

}

**private** **static** **void** themSinhVienDB(SinhVienDAO daodb, SinhVien sv) {

**boolean** result = daodb.insertSinhVien(sv);

System.***out***.println(result ? "Đã thêm vào DB" : "Lỗi thêm vào DB");

}

**private** **static** **void** xoaSinhVienDB(SinhVienDAO daodb, String maSV) **throws** IOException {

**boolean** result = daodb.deleteSinhVien(maSV);

System.***out***.println(result ? "Đã xóa từ DB" : "Lỗi xóa từ DB");

}

**private** **static** **void** xoaSinhVienTXT(SinhVienDAO daotxt, String maSV) **throws** IOException {

**boolean** result = daotxt.deleteSinhVien(maSV);

System.***out***.println(result ? "Đã xóa từ TXT" : "Lỗi xóa từ TXT");

}

* 1. **Nội dung hàm main trong class ChuongTrinh**

**public** **static** **void** main(String[] arg) {

**try** {

SinhVienFactory factory = **new** SinhVienFactory();

SinhVienDAO daotxt = factory.getPhuongThuc("txt");

SinhVienDAO daodb = factory.getPhuongThuc("not txt");

// Đọc và hiển thị

*hienThiDanhSach*(daotxt.readDanhSachSinhVien());

System.***out***.println("Sinh viên từ database");

*hienThiDanhSach*(daodb.readDanhSachSinhVien());

// Thêm sinh viên

*themSinhVienTXT*(daotxt, **new** SinhVien("T102002", "Huỳnh Quang", **true**, "2003/01/01"));

*themSinhVienDB*(daodb, **new** SinhVien("T101997", "Nguyễn Công", **true**, "1999-01-01"));

// Xóa sinh viên

*xoaSinhVienTXT*(daotxt, "T102002");

*xoaSinhVienDB*(daodb, "T102002");

} **catch** (Exception e) {e.printStackTrace(); }

}

}

1. **Nội dung file 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>simpleFactory</groupId>

<artifactId>simpleFactory</artifactId>

<version>0.0.1-SNAPSHOT</version>

<build>

<sourceDirectory>src</sourceDirectory>

<plugins>

<plugin>

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

<version>3.8.1</version>

<configuration>

<release>21</release>

</configuration>

</plugin>

</plugins>

</build>

<dependencies>

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<version>1.18.36</version>

<scope>provided</scope>

</dependency>

<!--

https://mvnrepository.com/artifact/com.microsoft.sqlserver/mssql-jdbc -->

<dependency>

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

<artifactId>mssql-jdbc</artifactId>

<version>12.8.1.jre11</version>

</dependency>

</dependencies>

</project>

1. **Nội dung file module-info.java**

**module** simpleFactory {

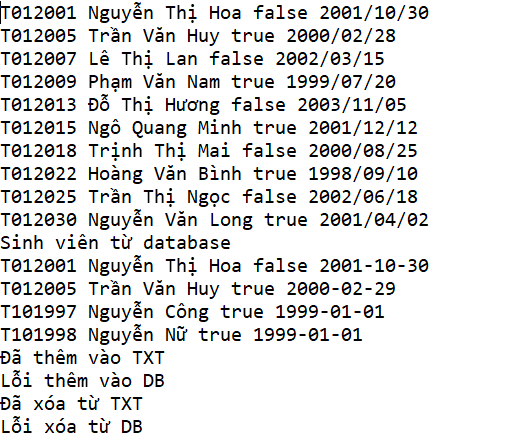
**exports** T1020625;

**requires** java.sql;

**requires** lombok;

}

1. **Hình minh hoạ chạy chương trình**

****

1. **Script của cơ sở dữ liệu**

CREATE DATABASE [ViDuDB1]

GO

USE [ViDuDB1]

GO

/\*\*\*\*\*\* Object: Table [dbo].[SinhVien] Script Date: 3/19/2025 5:15:06 PM \*\*\*\*\*\*/

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

INSERT [dbo].[SinhVien] ([MaSinhVien], [HoTen], [GioiTinhNam], [NgaySinh]) VALUES (N'T012001', N'Nguyễn Thị Hoa', 0, CAST(N'2001-10-30' AS Date))

GO

INSERT [dbo].[SinhVien] ([MaSinhVien], [HoTen], [GioiTinhNam], [NgaySinh]) VALUES (N'T012005', N'Trần Văn Huy', 1, CAST(N'2000-02-28' AS Date))

GO

INSERT [dbo].[SinhVien] ([MaSinhVien], [HoTen], [GioiTinhNam], [NgaySinh]) VALUES (N'T101997', N'Nguyễn Công', 1, CAST(N'1999-01-01' AS Date))

GO

INSERT [dbo].[SinhVien] ([MaSinhVien], [HoTen], [GioiTinhNam], [NgaySinh]) VALUES (N'T101998', N'Nguyễn Nữ', 1, CAST(N'1999-01-01' AS Date))

GO

1. **Minh hoạ dữ liệu nguồn dữ liệu text**

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

T012005,Trần Văn Huy,True,2000/02/28

T012007,Lê Thị Lan,False,2002/03/15

T012009,Phạm Văn Nam,True,1999/07/20

T012013,Đỗ Thị Hương,False,2003/11/05

T012015,Ngô Quang Minh,True,2001/12/12

T012018,Trịnh Thị Mai,False,2000/08/25

T012022,Hoàng Văn Bình,True,1998/09/10

T012025,Trần Thị Ngọc,False,2002/06/18

T012030,Nguyễn Văn Long,True,2001/04/02