import entidade.Aluno;

import helper.ArquivoHelper;

import interfaces.IAluno;

public class AlunoController implements IAluno {

@Override

public void criarAluno(Aluno aluno) {

aluno.setMatricula();

ArquivoHelper arquivoHelper = ArquivoHelper.getInstance();

arquivoHelper.gravarCSV(aluno);

}

@Override

public String lerAluno() {

ArquivoHelper arquivoHelper = ArquivoHelper.getInstance();

return arquivoHelper.lerCSV();

}

@Override

public void atualizarAluno(Aluno aluno) {

aluno.setMatricula();

ArquivoHelper arquivoHelper = ArquivoHelper.getInstance();

arquivoHelper.gravarCSV(aluno);

}

@Override

public boolean deletarAluno() {

ArquivoHelper arquivoHelper = ArquivoHelper.getInstance();

return arquivoHelper.deletarCSV();

}

}

////////////////////////////////////////////////////////////////////////////////////////////////////

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.time.LocalDateTime;

import java.util.Date;

import java.util.Random;

public class Aluno {

private String cpf;

private String nome;

private String curso;

private String matricula;

public String getCpf() {

return cpf;

}

public void setCpf(String cpf) {

this.cpf = cpf;

}

public String getNome() {

return nome;

}

public void setNome(String nome) {

this.nome = nome;

}

public String getCurso() {

return curso;

}

public void setCurso(String curso) {

this.curso = curso;

}

public String getMatricula() {

return matricula;

}

public void setMatricula() {

Date date = new Date();

DateFormat formatador = new SimpleDateFormat("YYYYMMddHHmmss");

String dataMatricula = formatador.format(date);

Random random = new Random();

Integer aleatorio = random.nextInt((99 - 1) + 1) + 1;

String codigo = String.format("%02d", aleatorio);

LocalDateTime dateTime = LocalDateTime.now();

int mes = dateTime.getMonthValue();

String semestre = "01";

if (mes > 6)

semestre = "02";

this.matricula = dataMatricula + "-" + codigo + "-" + semestre;

}

}

////////////////////////////////////////////////////////////////////////////////////////////////////

import java.util.Scanner;

import controller.AlunoController;

public class TestAluno {

public static void main(String[] args) {

int quantAluno;

String op, temp, conteudo;

Aluno aluno = new Aluno();

Scanner scanner = new Scanner(System.in);

AlunoController controller = new AlunoController();

//Menu de seleção

do {

System.out.println("\nMENU DE OPÇÕES\n");

System.out.println("1) Adicionar um novo aluno:");

System.out.println("2) Buscar alunos matriculados:");

System.out.println("3) Atualizar alunos no banco de dados:");

System.out.println("4) Deletar alunos do banco de dado:");

System.out.println("0) Finalizar");

op = scanner.nextLine();

switch (op) {

case "0":

System.out.println("\n\nFinalizando\n\n");

break;

case "1":

System.out.println("Adicionar a quantidade de alunos: ");

temp = scanner.nextLine();

quantAluno = Integer.parseInt(temp);

for (int i = 0; i < quantAluno; i++) {

aluno = new Aluno();

System.out.println("Informe o Nome: ");

aluno.setNome(scanner.nextLine());

System.out.println("Informe o CPF: ");

aluno.setCpf(scanner.nextLine());

System.out.println("Informe o Curso: ");

aluno.setCurso(scanner.nextLine());

controller.criarAluno(aluno);

System.out.println("\n\nAluno adicionado com sucesso!\n\n");

}

System.out.println("\n\nArquivo alunos criado com sucesso!\n\n");

break;

case "2":

conteudo = controller.lerAluno();

if (conteudo != null) {

System.out.println("\n\nArquivo alunos encontrado com sucesso:\n\n");

System.out.println(conteudo);

} else

System.out.println("\n\nArquivo não encontrado!\n\n");

break;

case "3":

if (controller.lerAluno() != null) {

controller.deletarAluno();

System.out.println("Quantos novos alunos deseja adicionar? ");

temp = scanner.nextLine();

quantAluno = Integer.parseInt(temp);

for (int i = 0; i < quantAluno; i++) {

aluno = new Aluno();

System.out.println("Informe o Nome: ");

aluno.setNome(scanner.nextLine());

System.out.println("Informe o CPF: ");

aluno.setCpf(scanner.nextLine());

System.out.println("Informe o Curso: ");

aluno.setCurso(scanner.nextLine());

controller.atualizarAluno(aluno);

System.out.println("\n\nAluno adicionado com sucesso!\n\n");

}

System.out.println("\n\nArquivo alunos alterado com sucesso!\n\n");

} else

System.out.println("\n\nArquivo não encontrado!\n\n");

break;

case "4":

if (controller.deletarAluno())

System.out.println("\n\nArquivo deletado com sucesso!\n\n");

else

System.out.println("\n\nArquivo não existente\n\n");

break;

default:

System.out.println("\n\nOpção invalida!\n\n");

}

}

while (!op.equals("0"));

}

}

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileWriter;

import java.io.IOException;

import java.nio.charset.StandardCharsets;

import entidade.Aluno;

public class ArquivoHelper {

private String caminho = "alunos.csv";

private String separador = ",";

private static ArquivoHelper instance = null;

private ArquivoHelper() {

}

public static ArquivoHelper getInstance() {

if (instance == null)

instance = new ArquivoHelper();

return instance;

}

public void gravarCSV(Aluno aluno) {

StringBuilder builder = new StringBuilder();

builder.append(aluno.getNome())

.append(separador)

.append(aluno.getCpf())

.append(separador)

.append(aluno.getCurso())

.append(separador)

.append(aluno.getMatricula());

try {

FileWriter writer = new FileWriter(caminho, true);

BufferedWriter bWriter = new BufferedWriter(writer);

//Gravando arquivo com dados formatados

bWriter.write(builder.toString());

bWriter.newLine();

bWriter.close();

} catch (IOException exception) {

exception.printStackTrace();

}

}

public String lerCSV() {

File file = new File(caminho);

try {

return FileUtils.readFileToString(file, StandardCharsets.UTF\_8);

} catch (FileNotFoundException exception) {

return null;

} catch (IOException exception) {

exception.printStackTrace();

}

return null;

}

public boolean deletarCSV() {

File file = new File(caminho);

return file.delete();

}

}

import entidade.Aluno;

public interface IAluno {

void criarAluno(Aluno aluno);

String lerAluno();

void atualizarAluno(Aluno aluno);

boolean deletarAluno();

}