Esercizio 1: Contatti telefonici

Scrivi un programma Java per gestire un elenco di contatti. Ogni contatto ha un nome e un numero di telefono. Il programma dovrebbe consentire all'utente di aggiungere nuovi contatti, rimuovere contatti esistenti, visualizzare l'elenco completo dei contatti e cercare un contatto per nome. Utilizza array per memorizzare i nomi dei contatti e i loro numeri di telefono.

Assicurati di strutturare il codice in modo da utilizzare metodi distinti per le operazioni principali (aggiunta, rimozione, visualizzazione e ricerca) e mantenere il codice ben organizzato e leggibile.

import java.util.Scanner;

public class GestioneContatti {

private static final int MAX\_CONTATTI = 100;

private String[] nomi;

private String[] numeri;

private int numContatti;

public GestioneContatti() {

nomi = new String[MAX\_CONTATTI];

numeri = new String[MAX\_CONTATTI];

numContatti = 0;

}

public void aggiungiContatto(String nome, String numero) {

if (numContatti < MAX\_CONTATTI) {

nomi[numContatti] = nome;

numeri[numContatti] = numero;

numContatti++;

System.out.println("Contatto aggiunto con successo.");

} else {

System.out.println("Limite massimo di contatti raggiunto.");

}

}

public void rimuoviContatto(String nome) {

int indice = -1;

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

if (nomi[i].equals (nome)) {

indice = i;

break;

}

}

if (indice != -1) {

for (int i = indice; i < numContatti - 1; i++) {

nomi[i] = nomi[i + 1];

numeri[i] = numeri[i + 1];

}

nomi[numContatti - 1] = null;

numeri[numContatti - 1] = null;

numContatti--;

System.out.println("Contatto rimosso con successo.");

} else {

System.out.println("Contatto non trovato.");

}

}

public void visualizzaElencoContatti() {

System.out.println("Elenco dei contatti:");

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

System.out.println(nomi[i] + ": " + numeri[i]);

}

}

public void cercaContatto(String nome) {

boolean trovato = false;

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

if (nomi[i].equals (nome)) {

System.out.println("Contatto trovato:");

System.out.println(nomi[i] + ": " + numeri[i]);

trovato = true;

break;

}

}

if (!trovato) {

System.out.println("Contatto non trovato.");

}

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

GestioneContatti gestore = new GestioneContatti();

while (true) {

System.out.println("\nMenu:");

System.out.println("1. Aggiungi contatto");

System.out.println("2. Rimuovi contatto");

System.out.println("3. Visualizza elenco contatti");

System.out.println("4. Cerca contatto per nome");

System.out.println("5. Esci");

System.out.print("Scelta: ");

int scelta = scanner.nextInt();

scanner.nextLine(); // Consuma il newline

switch (scelta) {

case 1:

System.out.print("Nome: ");

String nome = scanner.nextLine();

System.out.print("Numero di telefono: ");

String numero = scanner.nextLine();

gestore.aggiungiContatto(nome, numero);

break;

case 2:

System.out.print("Nome del contatto da rimuovere: ");

String nomeDaRimuovere = scanner.nextLine();

gestore.rimuoviContatto(nomeDaRimuovere);

break;

case 3:

gestore.visualizzaElencoContatti();

break;

case 4:

System.out.print("Nome del contatto da cercare: ");

String nomeDaCercare = scanner.nextLine();

gestore.cercaContatto(nomeDaCercare);

break;

case 5:

System.out.println("Arrivederci!");

System.exit(0);

default:

System.out.println("Scelta non valida. Riprova.");

}

}

}

}

Esercizio 2 – Gestione scolastica

Scrivi un programma Java per gestire un elenco di studenti e i loro voti in diverse materie. Il programma dovrebbe consentire all'utente di:

* Inserire i nomi degli studenti e i loro voti in diverse materie.
* Calcolare la media dei voti di ciascuno studente.
* Calcolare la media dei voti di ciascuna materia.
* Trovare il numero di studenti che hanno ottenuto un voto superiore a una determinata soglia.
* Visualizzare un elenco degli studenti che hanno ottenuto il voto massimo in una specifica materia.

import java.util.Scanner;

public class GestioneVoti {

private static final int NUM\_MATERIE = 3;

private static final int MAX\_STUDENTI = 50;

private static final int VOTO\_SOGLIA = 7;

private String[] nomiStudenti;

private int[][] votiStudenti;

private int numStudenti;

public GestioneVoti() {

nomiStudenti = new String[MAX\_STUDENTI];

votiStudenti = new int[MAX\_STUDENTI][NUM\_MATERIE];

numStudenti = 0;

}

public void inserisciVoti(String nome, int[] voti) {

if (numStudenti < MAX\_STUDENTI) {

nomiStudenti[numStudenti] = nome;

votiStudenti[numStudenti] = voti;

numStudenti++;

} else {

System.out.println("Limite massimo di studenti raggiunto.");

}

}

public double calcolaMediaVotiStudente(int indiceStudente) {

int[] voti = votiStudenti[indiceStudente];

double somma = 0;

for (int voto : voti) {

somma += voto;

}

return somma / NUM\_MATERIE;

}

public double calcolaMediaVotiMateria(int indiceMateria) {

double somma = 0;

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

somma += votiStudenti[i][indiceMateria];

}

return somma / numStudenti;

}

public int contaStudentiConVotoSuperioreASoglia(int soglia) {

int count = 0;

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

for (int voto : votiStudenti[i]) {

if (voto > soglia) {

count++;

break;

}

}

}

return count;

}

public void visualizzaStudentiConVotoMassimoInMateria(int indiceMateria) {

int votoMassimo = Integer.MIN\_VALUE;

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

if (votiStudenti[i][indiceMateria] > votoMassimo) {

votoMassimo = votiStudenti[i][indiceMateria];

}

}

System.out.println("Studenti con voto massimo in materia " + indiceMateria + ":");

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

if (votiStudenti[i][indiceMateria] == votoMassimo) {

System.out.println(nomiStudenti[i]);

}

}

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

GestioneVoti gestore = new GestioneVoti();

// Inserimento dei dati degli studenti

System.out.println("Inserisci i dati degli studenti:");

for (int i = 0; i < MAX\_STUDENTI; i++) {

System.out.println("Studente " + (i + 1) + ":");

System.out.print("Nome: ");

String nome = scanner.nextLine();

int[] voti = new int[NUM\_MATERIE];

for (int j = 0; j < NUM\_MATERIE; j++) {

System.out.print("Voto in materia " + (j + 1) + ": ");

voti[j] = scanner.nextInt();

}

scanner.nextLine(); // Consuma il newline

gestore.inserisciVoti(nome, voti);

System.out.println();

}

// Calcolo e visualizzazione della media dei voti di ciascuno studente

System.out.println("Media dei voti di ciascuno studente:");

for (int i = 0; i < gestore.numStudenti; i++) {

System.out.println(gestore.nomiStudenti[i] + ": " + gestore.calcolaMediaVotiStudente(i));

}

// Calcolo e visualizzazione della media dei voti di ciascuna materia

System.out.println("\nMedia dei voti di ciascuna materia:");

for (int i = 0; i < NUM\_MATERIE; i++) {

System.out.println("Materia " + (i + 1) + ": " + gestore.calcolaMediaVotiMateria(i));

}

// Conta studenti con voto superiore a una determinata soglia

System.out.println("\nNumero di studenti con voto superiore a " + VOTO\_SOGLIA + ": " +

gestore.contaStudentiConVotoSuperioreASoglia(VOTO\_SOGLIA));

// Visualizza studenti con voto massimo in una specifica materia

int materiaDaCercare = 0; // Indice della materia da cercare (0, 1 o 2)

gestore.visualizzaStudentiConVotoMassimoInMateria(materiaDaCercare);

scanner.close();

}

}