[Clase Estudiante](#_mue0u5ct9my)

[Clase main TareaSoftNotas](#_lkc9p38u9wns)

En el archivo adjunto en formato Zip, Rar se encuentra el código fuente para abrirse como proyecto en Netbeans.

# Clase Estudiante

//Edwin Trigos y Santiago Marquez B191 POO

package tareasoftnotas;

import static tareasoftnotas.Estudiante.cadena;

public class Estudiante {

String nombre;

String cedula;

float promedioAcumulado;

int creditosCursados;

public Estudiante(String nombre, String cedula, float promedioAcumulado, int creditosCursados) {

this.nombre = nombre;

this.cedula = cedula;

this.promedioAcumulado = promedioAcumulado;

this.creditosCursados = creditosCursados;

}

public String getNombre() {

return nombre;

}

public void setNombre(String nombre) {

this.nombre = nombre;

}

public String getCedula() {

return cedula;

}

public void setCedula(String cedula) {

this.cedula = cedula;

}

public float getPromedioAcumulado() {

return promedioAcumulado;

}

public void setPromedioAcumulado(float promedioAcumulado) {

this.promedioAcumulado = promedioAcumulado;

}

public int getCreditosCursados() {

return creditosCursados;

}

public void setCreditosCursados(int creditosCursados) {

this.creditosCursados = creditosCursados;

}

public float nuevoProm(float promedioAcumulado, int creditosVistos, float promedioSemestre, int creditosSemestre) {

float nuevoProm = (promedioAcumulado \* creditosVistos + promedioSemestre \* creditosSemestre) / (creditosVistos + creditosSemestre);

return nuevoProm;

}

public static String cadena(Estudiante[] estudiantes) {

String cadena = "";

for (int i = 0; i < estudiantes.length; i++) {

cadena += "\nCedula: " + estudiantes[i].getCedula() + "\n" + "Nombre: " + estudiantes[i].getNombre() + "\n" + "Promedio acumulado: " + estudiantes[i].getPromedioAcumulado() + "Creditos del estudiante: " + estudiantes[1].getCreditosCursados() + "\n";

}

return cadena;

}

//metodo que obtiene la lista de los estudiantes, sacando los valores de la matriz y anidandolos en un String

public static String getEstudiantesxpromedio(Estudiante[] estudiantes, float filtro) {

int cantidadEstudiantes = 0;

for (int i = 0; i < estudiantes.length;) {

if (estudiantes[i].getPromedioAcumulado() > filtro) {

cantidadEstudiantes++;

}

i++;

}

//arreglo para guardar los estudiantes que cumplan con el promedio

Estudiante[] estudiantesArreglo = new Estudiante[cantidadEstudiantes];

//agregando los estudiantes a la lista y devolciendolos como String

for (int i = 0; i < estudiantes.length;) {

if (estudiantes[i].getPromedioAcumulado() > filtro) {

for (int j = 0; j < 10; j++) {

//sacamos los datos del arreglo

String nom = estudiantes[i].getNombre();

String cedula = estudiantes[i].getCedula();

int creditos = estudiantes[i].getCreditosCursados();

float promedio = estudiantes[i].getPromedioAcumulado();

//creacion de estudiantes con el promedio

Estudiante e1 = new Estudiante(nom, cedula, promedio, creditos);

//insertando los estudiantes filtrados en el nuevo arreglo

estudiantesArreglo[i] = e1;

}

}

i++;

}

//retornando la lista con los estudiantes

String lista = cadena(estudiantesArreglo);

if (cantidadEstudiantes == 0) {

lista = "\n No se encontraron estudiantes con el promedio dado \n";

}

return lista;

}

public static void setNuevoPromedio(int i, String cedula, float promedioSemestre, int creditosVistos, Estudiante[] estudiantes) {

//calculo del nuevo promedio para el estudiante:

float nuevoprom = (estudiantes[i].getPromedioAcumulado() \* estudiantes[i].getCreditosCursados() + estudiantes[i].getPromedioAcumulado() \* promedioSemestre) / (creditosVistos + estudiantes[i].getCreditosCursados());

estudiantes[i].setPromedioAcumulado(nuevoprom);

}

public static String getEstudianteMejorProm(Estudiante[] estudiantes) {

Estudiante[] estudiantePepa = new Estudiante[1];

estudiantePepa[0] = estudiantes[1];

for (int i = 0; i < estudiantes.length;) {

if (estudiantes[i].getPromedioAcumulado() > estudiantePepa[0].getPromedioAcumulado()) {

estudiantes[i] = estudiantePepa[0];

}

i++;

}

//retornando la lista con los estudiantes

String informacionEstudiante = cadena(estudiantePepa);

if (estudiantePepa[0].getNombre() == null) {

informacionEstudiante = "\n No hay estudiantes en la lista \n";

}

return informacionEstudiante;

}

}

# Clase main TareaSoftNotas

//Edwin Trigos y Santiago Marquez B191 POO

package tareasoftnotas;

import java.util.Scanner;

import static tareasoftnotas.Estudiante.cadena;

import static tareasoftnotas.Estudiante.getEstudiantesxpromedio;

import static tareasoftnotas.Estudiante.setNuevoPromedio;

public class TareaSoftNotas {

public static void main(String[] args) {

//declaracion de Scanner

Scanner lector = new Scanner(System.in);

//variables para crear el objeto estudiantes

String nombre;

float promedio;

int creditos;

//declaracion de variables para usar en el nuevoProm del punto 2

//(se hace uso de las variable creditos y promedio ya creadas)

int opc;

String cedula = "";

//peticion del numero de estudiantes

int numeroEstudiantes;

System.out.println("Ingrese el numero de estudiantes a registrar: ");

numeroEstudiantes = lector.nextInt();

//creacion de la matriz de estudiantes

Estudiante[] estudiantes = new Estudiante[numeroEstudiantes];

//peticion de datos de acuerdo al numero de estudiantes ingresados

for (int i = 0; i < numeroEstudiantes;) {

int num = i + 1;

System.out.println("Nombre del estudiante " + num + " :");

nombre = lector.next();

System.out.println("Cedula del estudiante " + num + " :");

cedula = lector.next();

System.out.println("Promedio acumulado del estudiante " + num + " :");

promedio = lector.nextFloat();

System.out.println("Creditos cursados del estudiante " + num + " :");

creditos = lector.nextInt();

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

// Creacion de un nuevo objeto estudiante con los datos recolectados

Estudiante estudiante1 = new Estudiante(nombre, cedula, promedio, creditos);

// Asignacion del estudiante en una posicion de la matriz

estudiantes[i] = estudiante1;

i++;

}

boolean salir = false;

do {

//menu

System.out.println("-----------------------------Menu de opciones----------------------------- \n");

System.out.println("1. Calcular nuevo promedio: ");

System.out.println("2. listado de estudiantes con promedio: ");

System.out.println("3. listado de estudiantes actuales: ");

System.out.println("4. Buscar el estudiante con el mejor promedio y mostrar sus datos: ");

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

System.out.println("Seleccione una de las opciones anteriores: \n ");

opc = lector.nextInt();

switch(opc) {

case 1:

//peticion de la cedula del estudiantes

System.out.println("Cedula del estudiante a modificar el promedio: ");

cedula = lector.next();

for (int i = 0; i < estudiantes.length;) {

//hacer validacion con cedula

if (estudiantes[i].getCedula().equals(cedula)) {

System.out.println("Numero de creditos vistos en el semestre: ");

creditos = lector.nextInt();

System.out.println("Promedio obtenido por el estudiante en este semestre: ");

promedio = lector.nextFloat();

//se puede hacer la asignacion aqui o mediante un metodo

setNuevoPromedio(i, cedula, promedio, creditos, estudiantes);

i++;

}

}

break;

case 2:

if (opc == 2) {

System.out.println("Nota para filtrar el listado: ");

float filtro = lector.nextFloat();

String lista = getEstudiantesxpromedio(estudiantes, filtro);

System.out.println("\n------------------------------------ Estudiantes con el promedio superior a " + filtro + " ------------------------------------\n");

System.out.println(lista);

}

break;

case 3:

if (opc == 3) {

String listaMuestra = cadena(estudiantes);

System.out.println(listaMuestra);

}

break;

case 4:

if (opc == 4) {

String informacion = Estudiante.getEstudianteMejorProm(estudiantes);

System.out.println(informacion);

}

break;

case 0:

if (opc == 0) {

salir = true;

}

break;

default:

System.out.println("La opcion elegida es incorrecta");

}

} while (!salir);

}

}