Evidencia “DÍA 3 SEMANA 7” del 09/06:

1 - Avance del ejercicio anterior del 08/06.

**MAIN**

/\*\*

\*

\* @author Angelica

\*/

public class Electrodomesticos {

}

// creacion de metodos

class pubic {

public pubic() {

}

}

public static void main(String[] args) {

{

ArrayList<Electrodomesticos> electrodomestic = new ArrayList<>(0);

//Variables:

int opc = 0, tam = 0;

tam = Apoyo.tamArchivo(electrodomestic);

//Metodo while:

while(opc != 3) {

opc = Apoyo.menu();

switch(opc) {

case 1:

Apoyo.leerArchivo(electrodomestic);

break;

case 2:

Apoyo.mostrar(electrodomestic);

break;

switch(consumoEnergetico){

case 'A':

plus+=100;

break;

case 'B':

plus+=80;

break;

case 'C':

plus+=60;

break;

case 'D':

plus+=50;

break;

case 'E':

plus+=30;

break;

case 'F':

plus+=10;

break;

}

// creacion de objetos

Electrodomestico listaElectrodomesticos[]=new Electrodomestico[10];

listaElectrodomesticos[0]=new Electrodomestico(200, 60, 'C', "Verde");

listaElectrodomesticos[1]=new Lavadora(150, 30);

listaElectrodomesticos[2]=new Television(500, 80, 'E', "negro", 42, false);

listaElectrodomesticos[3]=new Electrodomestico();

listaElectrodomesticos[4]=new Electrodomestico(600, 20, 'D', "gris");

listaElectrodomesticos[5]=new Lavadora(300, 40, 'Z', "blanco", 40);

listaElectrodomesticos[6]=new Television(250, 70);

listaElectrodomesticos[7]=new Lavadora(400, 100, 'A', "verde", 15);

listaElectrodomesticos[8]=new Television(200, 60, 'C', "naranja", 30, true);

listaElectrodomesticos[9]=new Electrodomestico(50, 10);

}

}

**PADRE**

\*\*

\*

\* @author Angelica

\*/

public class padre {

private String color;

private char letraEnergia;

private double peso;

public padre(String color, char letraEnergia, double peso) {

this.color = color;

this.letraEnergia = letraEnergia;

this.peso = peso;

}

public padre() {

}

public void setColor(String color) {

this.color = color;

}

public void setLetraEnergia(char letraEnergia) {

this.letraEnergia = letraEnergia;

}

public String getColor() {

return color;

}

public char getLetraEnergia() {

return letraEnergia;

}

public double getPeso() {

return peso;

}

private void comprobarColor(String color){

//Colores disponibles

String colores[]={"blanco", "negro", "rojo", "azul", "gris"};

boolean encontrado=false;

for(int i=0;i<colores.length && !encontrado;i++){

if(colores[i].equals(color)){

encontrado=true;

}

}

if(encontrado){

this.color=color;

}else{

// consumo energetico

public void comprobarletraEnergia(char letraEnergia){

if(letraEnergia >=65 && consumoEnergetico<=70){

this.consumoEnergetico=consumoEnergetico;

}else{

}

}

/\*\*

**\*LAVADORA**

\* @author Angelica

\*/

public class Lavadora extends padre{

private int carga;

public Lavadora(String color, char letraEnergia, double peso){

super(color, letraEnergia, peso);

}

/\*\*

\*

\* @param carga

\*/

public void setCarga( boolean carga) {

}

public Lavadora(int carga) {

this.carga = carga;

}

//Metodo que muestre los datos:

public void mostrar(){

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

System.out.println("Tipo de Electrodomestico (1:Lavadora)");

System.out.println("Consumo Electrico: " + getLetraEnergia ());

System.out.println("Peso: " + getPeso() + "Kg");

System.out.println("Carga: " + carga + "frontal");

System.out.println();

}

}

/\*\*

**\*TELEVISOR**

\* @author Angelica

\*/

public class Televisor extends padre{

private int resolucion;

public Televisor(String color, char letraEnergia, double peso) {

super(color, letraEnergia, peso);

}

// CONSTRUCTOR

public Televisor(int resolucion, String color, char letraEnergia, double peso) {

super(color, letraEnergia, peso);

this.resolucion = resolucion;

}

public void mostrar(){

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

System.out.println("Tipo de Electrodomestico (2:Televisor)");

System.out.println("Consumo Electrico: " + getLetraEnergia());

System.out.println("Peso: " + getPeso() + "Kg");

System.out.println("Tamaño: " + resolucion + " Pixeles");

System.out.println();

}

}

/\*\*

\***RADIO**

\* @author Angelica

\*/

public class radio extends padre{

private String conexion;

private String tipo;

private boolean portatil;

public radio (String color, char letraEnergia, double peso){

super(color, letraEnergia, peso);

}

// constructor

public radio(String conexion, String tipo, boolean portatil) {

this.conexion = conexion;

this.tipo = tipo;

this.portatil = portatil;

}public void mostrar(){

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

System.out.println("Tipo de Electrodomestico (3:radio)");

System.out.println("Consumo Electrico: " + getLetraEnergia());

System.out.println("Peso: " + getPeso() + "Kg");

System.out.println("conexion: " + " bluetooth");

System.out.println("tipo: " + " minicomponente");

System.out.println("portatil: " +" bluetooth");

System.out.println();

2 - Crear una superclase Persona de tipo abstracto con sus respectivos atributos y métodos propios de la clase y métodos abstractos, luego a partir de esta clase crear como mínimo tres subclases.

**CLASES PRINCIPAL**

/\*\*

\*

\* @author Angelica

\*/

public class clasePrincipal {

public static void main(String[] args) {

//Creamos el objeto

Aula aula=new Aula();

//Indicamos si se puede dar la clase

if(aula.darClase()){

aula.notas();

}

PERSONA

\*\*

\*

\* @author Angelica

\*/

// atributos clase padre

public class Persona {

private static class MetodosSueltos {

public MetodosSueltos() {

}

}

public abstract class estudiante {

private String nombre;

private char sexo;

private int edad;

public estudiante() {

}

public estudiante(String nombre, char sexo, int edad) {

this.nombre = nombre;

this.sexo = sexo;

this.edad = edad;

}

public String getNombre() {

return nombre;

}

public char getSexo() {

return sexo;

}

public int getEdad() {

return edad;

}

public void setNombre(String nombre) {

this.nombre = nombre;

}

public void setSexo(char sexo) {

this.sexo = sexo;

}

public void setEdad(int edad) {

this.edad = edad;

}

private final String[] NOMBRES\_CHICOS={"Jose", "Fernando", "Alberto", "Fernando", "Oscar"};

private final String[] NOMBRES\_CHICAS={"Maria", "Laura", "Claudia", "Consuelo", "Elena"};

private final int CHICO=0;

private final int CHICA=1;

//entre 0 y 1

int determinar\_sexo=MetodosSueltos.generaNumeroAleatorio(0,1);

//Si es 0 es un chico

if(determinar\_sexo==CHICO){

nombre=NOMBRES\_CHICOS[MetodosSueltos.generaNumeroAleatorio(0,4)];

sexo='H';

}else{

nombre=NOMBRES\_CHICAS[MetodosSueltos.generaNumeroAleatorio(0,4)];

sexo='M';

}

disponibildiad();

public boolean isAsistencia() {

return asistencia;

}

public void setAsistencia(boolean asistencia) {

this.asistencia = asistencia;

}

//abtracto, las clases hijas deben implementarlo

public abstract void disponibilidad();

**ESTUDIANTE**

/\*\*

\*

\* @author Angelica

\*\*/

}

public class estudiante extends Persona {

private boolean asistencia;

private int codigoEstudiante;

private float notaFinal;

public estudiante(boolean asistencia, int codigoEstudiante, float notaFinal) {

this.asistencia = asistencia;

this.codigoEstudiante = codigoEstudiante;

this.notaFinal = notaFinal;

public estudiante() {

public estudiante (String "nombre",chard 'sexo',int edad, booleano asistencia, codigoEstudiante, notaFinal){

super (nombre,sexo, edad);

nota=MetodosSueltos.generaNumeroAleatorio(0,10);

super.setEdad(MetodosSueltos.generaNumeroAleatorio(12,15));

public float getnotaFinal {

return notaFinal;

public void disponibilidad() {

int prob=MetodosSueltos.generaNumeroAleatorio(0, 100);

if(prob<50){

super.setAsistencia(false);

}else{

super.setAsistencia(true);

}

**PROFESOR**

/\*\*

\*

\* @author Angelica

\*/

public class proefesor {

public class proefesor extends Persona{

/\*Atributos\*/

private String materia;

/\*Constructores\*/

public proefesor(){

super(); //Llama al constructor padre

super.setEdad(MetodosSueltos.generaNumeroAleatorio(25,50)); //llama al metodo padre

materia=Constantes.MATERIAS[MetodosSueltos.generaNumeroAleatorio(0,2)];

}

/\*Metodos\*/

/\*\*

\* Devuelve la materia del profesor

\* @return

\*/

public String getMateria() {

return materia;

}

/\*\*

\* Modifica la materia del profesor

\* @param materia

\*/

public void setMateria(String materia) {

this.materia = materia;

}

/\*\*

\* Calcula la disponibilidad del profesor(20%)

\*/

@Override

public void disponibilidad() {

int prob=MetodosSueltos.generaNumeroAleatorio(0, 100);

if(prob<20){

super.setAsistencia(false);

}else{

super.setAsistencia(true);

}

**AULA**

\*\*

\*

\* @author Angelica

\*/

public class Aula {

private int id;

private proefesor profesor;

private estudiante[] alumnos;

private String materia;

/\*Constantes\*/

private final int MAX\_ALUMNOS=20;

/\*Constructores\*/

public Aula(){

id=1;

profesor=new proefesor();

alumnos= new estudiante[MAX\_ALUMNOS];

creaAlumnos();

materia=Constantes.MATERIAS[MetodosSueltos.generaNumeroAleatorio(0,2)];

}

/\*Metodos\*/

/\*\*

\* Crea los alumnos para el aula

\*/

private void creaAlumnos(){

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

alumnos[i]=new Alumno();

}

}

/\*\*

\* Indica si la asistencia de los alumnos es mayor del 50%

\* @return

\*/

private boolean asistenciaAlumnos(){

int cuentaAsistencias=0;

//contamos las asistencias

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

if(alumnos[i].isAsistencia()){

cuentaAsistencias++;

}

}

//Muestro la asistencia total

System.out.println("Hay "+cuentaAsistencias+" alumnos");

return cuentaAsistencias>=((int)(alumnos.length/2));

}

/\*\*

\* Indicamos si se puede dar clase

\* @return

\*/

public boolean darClase(){

//Indicamos las condiciones para que se pueda dar la clase

if(!profesor.isAsistencia()){

System.out.println("El profesor no esta, no se puede dar clase");

return false;

}else if(!profesor.getMateria().equals(materia)){

System.out.println("La materia del profesor y del aula no es la misma, no se puede dar clase");

return false;

}else if (!asistenciaAlumnos()){

System.out.println("La asistencia no es suficiente, no se puede dar clase");

return false;

}

System.out.println("Se puede dar clase");

return true;

}

/\*\*

\* Indicamos las notas de los alumnos aprobados, chicos y chicas

\*/

public void notas(){

int chicosApro=0;

int chicasApro=0;

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

//Comprobamos si el alumno esta aprobado

if(alumnos[i].getNota()>=5){

//Segun el sexo, aumentara uno o otro

if(alumnos[i].getSexo()=='H'){

chicosApro++;

}else{

chicasApro++;

}

System.out.println(alumnos[i].toString());

System.out.println("Hay "+chicosApro+" chicos y "+chicasApro+" chicas aprobados/as");

}

}