Evidencia "DÍA 4 SEMANA 6" del 01/06:

1 - Avance del ejercicio anterior del 30/05 si no lo habían terminad

CONTINUACIÓN EJERCICIO DE LA CLASE ANTERIOR………….

CORREGIDO ERROR

/\*\*

\*

\* @author Angelica

\*/

**public class Personas** {

private String nombre;

private int edad;

private double estatura;

private char sexo;

// metodo permite crear objetos -Metodo por defecto

public Personas ()

{

this.nombre="maria";

this.edad=25;

this.estatura=1.60;

this.sexo= 'f';

// constructor con parametros

}

public Personas(String nombre, int edad, double estatura, char sexo) {

this.nombre = nombre;

this.edad = edad;

this.estatura = estatura;

this.sexo = sexo;

}

//accesadores= permiten accesar al ultimo valor de la clase

public String getNombre()

{

return nombre;

}

public int getEdad()

{

return edad;

}

public double getEstatura()

{

return estatura;

}

public char getSexo()

{return sexo;

}

//mutadores== cambia los valores de las variables se usan como reglas de negocio validadciones

public void setNombre(String nombre)

{

this.nombre=nombre;

}

public void setEdad(int edad)

{

this.estatura=estatura;

}

public void setSexo(char sexo)

{ this.sexo=sexo;

}

{

if ((sexo == 'm')|| (sexo=='f')){

this.sexo=sexo;

}else {

System.out.println("formato de sexo no valido");

}

}

// metodo "customer" para imprimir datos de la clase

public void imprimirPersonas(){

System.out.println("\*\*\*\*\*\*\*\* DATOS DE PERSONAS");

System.out.println("Su nombte es:"+this.getNombre());

System.out.println("Su edad es: "+this.edad);

System.out.println("Su estatura es:"+this.estatura);

if (this.getSexo()=='m'){

System.out.println("Su sexo es masculino");

}else if (this.getSexo()=='f'){

System.out.println("Su sexo es femenino");

}else{

System.out.println("Su formato no es validad");

}

{

System.out.println("Su sexo es femenino");

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

System.out.println("");

}

**test persona**

/\*\*

\*

\* @author Angelica

\*/

public class TestPersona {

public static void main (String[]arg){

Personas objPersona1 = new Personas();

//System.out.println("Su nombre es:"+objPersona1.getNombre());

Personas objPersona2= new Personas("Pedro", 29, 1.70, 'm') ;

Personas objPersona3=new Personas( "Juanita", 20, 1.65, 'f');

//System.out.println("Su nombre del objeto 2 es:"+objPersona2.getNombre());

objPersona1.imprimirPersonas();

objPersona2.imprimirPersonas();

objPersona3.imprimirPersonas();

}

2.- Crea una clase llamada Contador que contenga un único atributo entero llamado cont. La clase tendrá los siguientes constructores: Constructor por defecto, Constructor con parámetros para inicializar el contador con un valor no negativo. Si el valor inicial que se recibe es negativo el contador tomará el valor cero como valor inicial. Además de los métodos getter y setter, la clase contendrá los métodos:

- incrementar: incrementa el contador en una unidad;

- decrementar: decrementa el contador en una unidad.

El contador nunca podrá tener un valor negativo. Si al decrementar se alcanza un valor negativo el contador toma el valor cero. Una vez creada la clase escribe un método main para probar la clase.

**Clase contador**

package clasecontador;

/\*\*

\*

\* @author Angelica

\*/

public class Contador {

private int cont;

//constructor por defecto

public Contador() {

}

//constructor con parámetros

public Contador(int cont) {

if (cont < 0) {

this.cont = 0;

} else {

this.cont = cont;

}

}

//constructor copia

public Contador(final Contador c) {

cont = c.cont;

}

//getter

public int getCont() {

return cont;

}

//setter

public void setCont(int cont) {

if (cont < 0) {

this.cont = 0;

} else {

this.cont = cont;

}

}

//método incrementar contador

public void incrementar() {

cont++;

}

//método decrementar contador

public void decrementar() {

cont--;

if (cont < 0) {

cont = 0;

}

}

}//Fin de la clase Contador

**MAIN CLASE CONTADOR**

import clasecontador.Contador;

import java.util.Scanner;

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

/\*\*

\*

\* @author Angelica

\*/

public class NewClass {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

//Utilizar el constructor por defecto

Contador contador1 = new Contador();

int n;

System.out.println("Introduce valor para inicializar el contador: ");

n = sc.nextInt();

//asignar un valor al contador

contador1.setCont(n);

//incrementar el contador

contador1.incrementar();

//mostrar el valor actual

System.out.println(contador1.getCont());

contador1.incrementar();

contador1.incrementar();

//mostrar el valor actual

System.out.println(contador1.getCont());

//restar 1 al valor del contador

contador1.decrementar();

//mostrar el valor actual

System.out.println(contador1.getCont());

//crear un nuevo objeto Contador con valor inicial 10

Contador contador2 = new Contador(10);

//incrementar y decrementar el contador2 y mostrar su valor

contador2.incrementar();

System.out.println(contador2.getCont());

contador2.decrementar();

System.out.println(contador2.getCont());

//crear un objeto Contador utilizando el constructor copia

//se crea el objeto contador3 como copia de contador2

Contador contador3 = new Contador(contador2);

//mostrar el valor de contador3

System.out.println(contador3.getCont());

}

}