

EJEMPLO DE APLICACIÓN HERENCIA - POLIMORFISMO

CASO DE NEGOCIO:

Una empresa de servicios requiere una aplicación informática para administrar los datos de su personal.

Del personal se conoce: número de DNI, apellidos y fecha de ingreso (**día, mes, año**).

Existen dos categorías de personal: el **personal contratado** y el **personal a destajo**.

Los empleados contratados tienen un sueldo básico y un porcentaje adicional en función al tiempo laborado:

De 0 a 3 años: 5%

De 4 a 7 años: 10%

De 8 a 15 años: 15%

Y más de 15 años: 20%

Los empleados a destajo tienen un número de clientes captados y un monto por cliente captado (valor fijo s/. 130.0).

Maneje el siguiente menú de opciones:

Menú de Opciones

1. Registrar empleado contratado
2. Registrar empleado por destajo
3. Mostrar salarios por empleado.
4. Salir

SOLUCIÓN

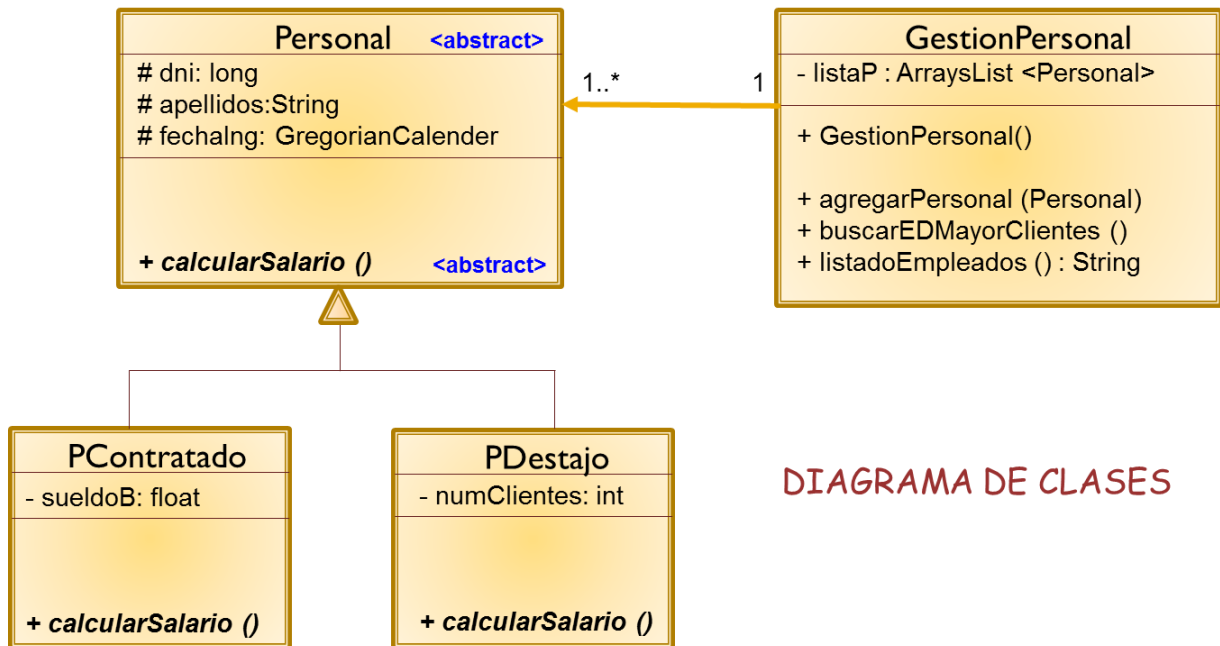
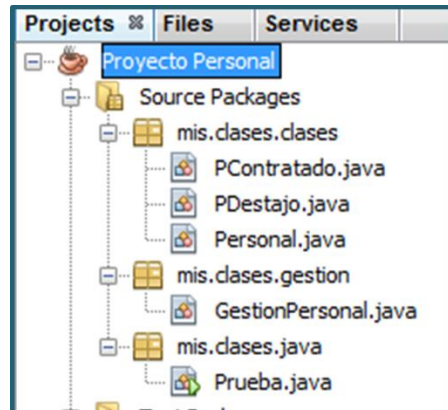


DIAGRAMA DE CLASES

Considera la siguiente organización de los paquetes:



Diseño de la clase abstracta **Personal**:

```
1  package mis.clases.clases;
2
3  import java.util.Calendar;
4  import java.util.GregorianCalendar;
5
6  public abstract class Personal {
7      //atributos
8      protected String dni;
9      protected String apellidos;
10     protected GregorianCalendar fechaIngreso;
11
12     //constructor
13     public Personal(String dni, String apellidos, GregorianCalendar fechaIngreso) {
14         this.dni = dni;
15         this.apellidos = apellidos;
16         this.fechaIngreso = fechaIngreso;
17     }
18
19     //getters y setters
20     public String getDni() { ...3 lines }
21
22     public void setDni(String dni) { ...3 lines }
23
24     public String getApellidos() { ...3 lines }
25
26     public void setApellidos(String apellidos) { ...3 lines }
27
28     public GregorianCalendar getFechaIngreso() { ...3 lines }
29
30     public void setFechaIngreso(GregorianCalendar fechaIngreso) { ...3 lines }
31
32
33
34
35
```

```

39 //override
40 public String toString(){
41     int dd,mm,aa;
42
43     dd=fechaIngreso.get(Calendar.DAY_OF_MONTH);
44     mm=fechaIngreso.get(Calendar.MONTH)+1;
45     aa=fechaIngreso.get(Calendar.YEAR);
46
47     return "\nDNI : " + dni + "\nApellidos : "+apellidos +
48         "\nFecha de Ingreso : " + dd+"/"+mm + "/" + aa;
49 }
50
51 //método abstracto
52 public abstract float calcularSalario();
53 }

```

Diseño de la subclase **PContratado**:

```

1  package mis.clases.clases;
2  import java.util.Calendar;
3  import java.util.GregorianCalendar;
4
5  public class PContratado extends Personal{
6      private float sueldoB;
7
8      public PContratado(float sueldoB, String dni, String apellidos,
9          GregorianCalendar fechaIngreso) {
10         super(dni, apellidos, fechaIngreso);
11         this.sueldoB = sueldoB;
12     }
13     public float getSueldoB() {...3 lines }
14     public void setSueldoB(float sueldoB) {...3 lines }
15
16
17
18
19
20 //override
21 public String toString(){
22     return super.toString() + "\nSalario : " + calcularSalario();
23 }

```

```

21 public float calcularSalario(){
22     float salario;
23     GregorianCalendar factual=new GregorianCalendar();
24     int añosTrans=factual.get(Calendar.YEAR)-getFechaIngreso().get(Calendar.YEAR);
25     if(getFechaIngreso().get(Calendar.MONTH)>factual.get(Calendar.MONTH) ||
26         getFechaIngreso().get(Calendar.MONTH)==factual.get(Calendar.MONTH) &&
27         getFechaIngreso().get(Calendar.DAY_OF_MONTH)>factual.get(Calendar.DAY_OF_MONTH)){
28         añosTrans--;
29     }
30     if(añosTrans<=3){
31         salario=1.05f*sueldoB;
32     }else if(añosTrans>3 && añosTrans<=7){
33         salario=1.1f*sueldoB;
34     }else if(añosTrans>7 && añosTrans<=15){
35         salario=1.15f*sueldoB;
36     }else{
37         salario=1.20f*sueldoB;
38     }
39     return salario;
40 }

```

Diseño de la subclase **PDestajo**:

```

1 package mis.clases.clases;
2
3 import java.util.GregorianCalendar;
4
5 public class PDestajo extends Personal{
6     //atributos
7     private static final float SALARIO=130.0f;
8     private int cantClientes;
9
10    //constructor
11    public PDestajo(int cantClientes, String dni, String apellidos,
12        GregorianCalendar fechaIngreso) {
13        super(dni, apellidos, fechaIngreso);
14        this.cantClientes = cantClientes;
15    }
16    public int getCantClientes() { ...3 lines }
17
18
19
20    //override
21    public String toString(){
22        return super.toString() + "\nSalario : " + calcularSalario();
23    }
24
25    public float calcularSalario(){
26        return cantClientes*SALARIO;
27    }
28 }

```

Diseño de la subclase **GestionPersonal**:

```
1  package mis.clases.gestion;
2
3  import java.util.ArrayList;
4  import mis.clases.clases.PDestajo;
5  import mis.clases.clases.Personal;
6
7  public class GestionPersonal {
8      private ArrayList<Personal> listaP;
9
10     public GestionPersonal() {
11         listaP=new ArrayList<Personal>();
12     }
13
14     public void agregarPersonal(Personal refP){
15         listaP.add(refP);
16     }
17
18     public String listadoEmpleado(){
19         String cad=" ";
20
21         for(int i=0;i<listaP.size();i++){
22             cad+="\n\tEmpleado " + (i+1) + ": " + listaP.get(i);
23         }
24         return cad;
25     }
26
27     public void buscarEDMayorClientes(){
28
29         int mayor=0;
30         PDestajo ref=null;
31         String nombreED="";
32         for(int i=0;i<listaP.size();i++){
33             if(listaP.get(i) instanceof PDestajo){
34                 ref=(PDestajo)listaP.get(i);
35                 if(ref.getCantClientes()>mayor){
36                     mayor=ref.getCantClientes();
37                     nombreED=ref.getApellidos();
38                 }
39             }
40         }
41         System.out.println(" Datos del empleado : " + nombreED);
42     }
43 }
44
```


Diseño de la subclase **Prueba**:

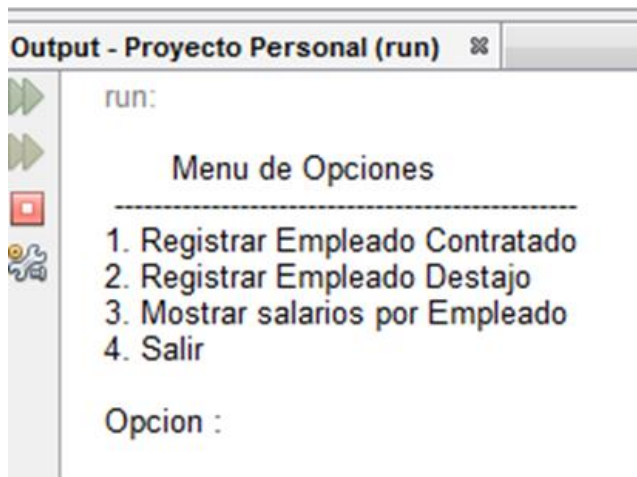
```
1 package mis.clases.java;
2
3 import java.util.GregorianCalendar;
4 import java.util.Scanner;
5 import mis.clases.clases.PContratado;
6 import mis.clases.clases.PDestajo;
7 import mis.clases.clases.Personal;
8 import mis.clases.gestion.GestionPersonal;
9
10 public class Prueba {
11     public static void main(String[] args) {
12         GestionPersonal refG=new GestionPersonal();
13
14         String dni="", ape="";
15         int op,dd,mm,aa;
16         GregorianCalendar fIngreso=null;
17
18         Scanner sc=new Scanner(System.in);
19         do{
20             System.out.println("\n      Menu de Opciones");
21             System.out.println("-----");
22             System.out.print("1. Registrar Empleado Contratado \n");
23             System.out.print("2. Registrar Empleado Destajo \n");
24             System.out.print("3. Mostrar salarios por Empleado \n");
25             System.out.print("4. Salir \n");
26             System.out.print("\nOpcion : ");
27             op=sc.nextInt();
28
29             if(op == 1 || op==2){
30                 System.out.print("Ingrese DNI : ");
31                 dni=sc.next();
32                 System.out.print("Ingrese apellidos : ");
33                 ape=sc.next();
34                 System.out.print("Dia : ");
35                 dd=sc.nextInt();
36                 System.out.print("Mes [1-12] : ");
37                 mm=sc.nextInt();
38                 System.out.print("Año : ");
39                 aa=sc.nextInt();
40                 fIngreso=new GregorianCalendar(aa, mm-1, dd);
41             }
42         }
```

```

42     if(op==1){
43         System.out.print("Ingrese sueldo base : ");
44         float sb=sc.nextFloat();
45
46         Personal refPC=new PContratado(sb, dni, ape, fIngreso);
47         refG.agregarPersonal(refPC);
48
49     }else if(op==2){
50         System.out.print("Ingrese cantidad de clientes : ");
51         int cantC=sc.nextInt();
52
53         Personal refPD=new PDestajo(cantC, dni, ape, fIngreso);
54         refG.agregarPersonal(refPD);
55
56     }else if(op==3){
57         System.out.println("\n\n LISTADO DE EMPLEADOS " + refG.listadoEmpleado());
58     }else if(op==4){
59         System.out.println("\nEl programa ha terminado ");
60     }else{
61         System.out.println("\nError ... opcion incorrecta");
62     }
63 }while(op!=4);
64
65 refG.buscarEDMayorClientes();
66 }
67 }

```

EJECUTAR:



Tarea:
Diseñe una clase de prueba con
interfaz gráfica.