

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE CIENCIAS QUÍMICA E INGENIERÍAS



Prof. Mayra Duran Rodríguez

Programación orientada a objetos

Practica #8

Mario Mendoza Virgen

Matricula 1261594



Tijuana, Baja California, México martes 1 de diciembre del 2020

```

MENU PRINCIPAL
1) INGRESAR EMPLEADOS ASALARIADOS
2) INGRESAR EMPLEADOS POR HORAS
3) INGRESAR EMPLEADOS POR COMISION
4) INGRESAR EMPLEADOS ASALARIADOS POR COMISION
5) MOSTRAR NOMINA DE TODOS LOS EMPLEADOS
6) SALIR DEL PROGRAMA
SELECCIONA UN OPCION: 1

Nombre: Juan

ID: 2360

Horas trabajadas: 50
Nomina de la semana: 1255.25
Felicidades, este mes se les dara 10% mas, total: 1380.775

```

```

MENU PRINCIPAL
1) INGRESAR EMPLEADOS ASALARIADOS
2) INGRESAR EMPLEADOS POR HORAS
3) INGRESAR EMPLEADOS POR COMISION
4) INGRESAR EMPLEADOS ASALARIADOS POR COMISION
5) MOSTRAR NOMINA DE TODOS LOS EMPLEADOS
6) SALIR DEL PROGRAMA
SELECCIONA UN OPCION: 2

Nombre: Pedro

ID: 9084

Horas trabajadas: 42
Nomina de la semana con horas extra: 4924.24

```

```

MENU PRINCIPAL
1) INGRESAR EMPLEADOS ASALARIADOS
2) INGRESAR EMPLEADOS POR HORAS
3) INGRESAR EMPLEADOS POR COMISION
4) INGRESAR EMPLEADOS ASALARIADOS POR COMISION
5) MOSTRAR NOMINA DE TODOS LOS EMPLEADOS
6) SALIR DEL PROGRAMA
SELECCIONA UN OPCION: 3

Nombre: Luis

ID: 96321

Horas trabajadas: 56

Ingresa la cantidad en $ total de vetas: 1053.23

Comision por venta del 0.15%
Nomina basada en ventas: 157.9845

MENU PRINCIPAL
1) INGRESAR EMPLEADOS ASALARIADOS
2) INGRESAR EMPLEADOS POR HORAS
3) INGRESAR EMPLEADOS POR COMISION
4) INGRESAR EMPLEADOS ASALARIADOS POR COMISION
5) MOSTRAR NOMINA DE TODOS LOS EMPLEADOS
6) SALIR DEL PROGRAMA
SELECCIONA UN OPCION: 4

Nombre: Raul

ID: 7485

Horas trabajadas: 35

Ingresa la cantidad en $ total de vetas: 11625
Salario Base: 1255.25 + Comisiones: 1743.75

Nomina: 2999.0

```

```
MENU PRINCIPAL
1) INGRESAR EMPLEADOS ASALARIADOS
2) INGRESAR EMPLEADOS POR HORAS
3) INGRESAR EMPLEADOS POR COMISION
4) INGRESAR EMPLEADOS ASALARIADOS POR COMISION
5) MOSTRAR NOMINA DE TODOS LOS EMPLEADOS
6) SALIR DEL PROGRAMA
SELECCIONA UN OPCION: 5
```

```
Empleado
Nombre: JUAN
ID: 2360
Horas: 50
Nomina: 1380.775
```

```
Empleado
Nombre: PEDRO
ID: 9084
Horas: 42
Nomina: 4924.24
```

```
Empleado
Nombre: LUIS
ID: 96321
Horas: 56
Ventas: 1053.23
Nomina: 157.9845
```

```
Empleado
Nombre: RAUL
ID: 7485
Horas: 35
Ventas: 11625.0
Nomina: 2999.0
```

```
MENU PRINCIPAL
1) INGRESAR EMPLEADOS ASALARIADOS
2) INGRESAR EMPLEADOS POR HORAS
3) INGRESAR EMPLEADOS POR COMISION
4) INGRESAR EMPLEADOS ASALARIADOS POR COMISION
5) MOSTRAR NOMINA DE TODOS LOS EMPLEADOS
6) SALIR DEL PROGRAMA
SELECCIONA UN OPCION: 6
```

```
Saliendo del programa...
PS C:\Users\mario> █
```



```

        case '6':
            System.out.println("\nSaliendo del programa...");
            break;

        default:
            System.out.println("Error, Opcion no encontrada, intentelo n
uevamente \n");
            break;
    }

    }while(opc != '6');

    sc.close();
}

public static void menuPrincipal(){
    System.out.println("\tMENU PRINCIPAL");
    System.out.println("1) INGRESAR EMPLEADOS ASALARIADOS");
    System.out.println("2) INGRESAR EMPLEADOS POR HORAS");
    System.out.println("3) INGRESAR EMPLEADOS POR COMISION");
    System.out.println("4) INGRESAR EMPLEADOS ASALARIADOS POR COMISION")
;

    System.out.println("5) MOSTRAR NOMINA DE TODOS LOS EMPLEADOS");
    System.out.println("6) SALIR DEL PROGRAMA");
    System.out.print("SELECCIONA UN OPCION: ");
}
}

```

Interfaz.java

```
public interface Interfaz {  
  
    public void name(String Name);  
    public void id(int ID);  
    public void horas(int Horas);  
    public void salario(double salario);  
    public void ventas(double ventas);  
  
}
```

Empleados.java

```
public abstract class Empleados implements Interfaz{  
    String Name;  
    int ID;  
    int Horas;  
    double Salario;  
    double Ventas;  
  
    public Empleados(String Name, int ID, int Horas, double Salario, double V  
entas) {  
        this.Name = Name;  
        this.ID = ID;  
        this.Horas = Horas;  
        this.Salario = Salario;  
        this.Ventas = Ventas;  
    }  
  
    public void name(){}  
    public void id(){}  
    public void horas(){}  
    public void salario(){}  
    public void ventas(){}  
  
    protected abstract void generaEmpleadoFijo(Empleados P0[],int N);  
    protected abstract void ingresaNombre(Empleados P0[],int N);  
    protected abstract void ingresaID(Empleados P0[],int N);  
    protected abstract void ingresaHoras(Empleados P0[],int N);  
    protected abstract void ingresaSalario(Empleados P0[],int N);  
    protected abstract void ingresaVentas(Empleados P0[],int N);  
  
}
```

EmpleadoFijo.java

```
import java.util.*;

public class EmpleadoFijo extends Empleados{

    public static final double S_Base = 1255.25;
    static Scanner sc = new Scanner(System.in);

    public EmpleadoFijo(String Name, int ID, int Horas, double Salario, double Ventas) {
        super(Name, ID, Horas, Salario, Ventas);
    }

    public void name(String Name){
        this.Name = Name;
    }

    public void id(int ID){
        this.ID = ID;
    }

    public void horas(int Horas){
        this.Horas = Horas;
    }

    public void salario(double Salario){
        this.Salario = Salario;
    }

    public void ventas(double Ventas){
        this.Ventas = Ventas;
    }

    @Override
    protected void ingresaNombre(Empleados[] P0, int N) {
        System.out.print("\nNombre: ");
        name(sc.nextLine().toUpperCase());
    }

    @Override
    protected void ingresaID(Empleados[] P0, int N) {
        System.out.print("\nID: ");
        id(sc.nextInt());
    }

    @Override
```

```
protected void ingresaHoras(Empleados[] P0, int N) {  
    System.out.print("\nHoras trabajadas: ");  
    horas(sc.nextInt());  
}  
  
@Override  
protected void ingresaSalario(Empleados[] P0, int N) {  
    System.out.println("Nomina de la semana: "+S_Base);  
  
    System.out.println("Felicidades, este mes se les dara 10% mas, total  
: "+(S_Base*1.1) + "\n");  
    salario(S_Base*1.1);//1.1 es el 10% extra del mes  
    sc.nextLine();  
}  
  
}
```


EmpleadoxHoras.java

```
import java.util.*;

public class EmpleadoxHoras extends Empleados{

    public static final double Xhoras = 120.25;//Pago por hora de trabajo
    public static final double extraHoras = 57.12;//Pago por hora extra

    static Scanner sc = new Scanner(System.in);

    public EmpleadoxHoras(String Name, int ID, int Horas, double Salario, double Ventas) {
        super(Name, ID, Horas, Salario, Ventas);
    }

    public void name(String Name){
        this.Name = Name;
    }

    public void id(int ID){
        this.ID = ID;
    }

    public void horas(int Horas){
        this.Horas = Horas;
    }

    public void salario(double Salario){
        this.Salario = Salario;
    }

    public void ventas(double Ventas){
        this.Ventas = Ventas;
    }

    @Override
    protected void ingresaNombre(Empleados[] P0, int N) {
        System.out.print("\nNombre: ");
        name(sc.nextLine().toUpperCase());
    }

    @Override
    protected void ingresaID(Empleados[] P0, int N) {
        System.out.print("\nID: ");
        id(sc.nextInt());
    }
}
```

```

@Override
protected void ingresaHoras(Empleados[] P0, int N) {
    System.out.print("\nHoras trabajadas: ");
    horas(sc.nextInt());
}

@Override
protected void ingresaSalario(Empleados[] P0, int N) {
    double sFinal = 0;

    if(P0[N].Horas <= 40){
        sFinal = P0[N].Horas * Xhoras;
        System.out.println("Nomina de la semana: " + sFinal + "\n");
    }

    else{
        sFinal = (40 * Xhoras + (P0[N].Horas-40)*extraHoras);
        System.out.println("Nomina de la semana con horas extra: "+ sFin
al + "\n");
    }

    salario(sFinal);
    sc.nextLine();
}
}

```

EmpleadoxComision.java

```
import java.util.*;

public class EmpleadoxComision extends Empleados{

    static Scanner sc = new Scanner(System.in);
    public static final double comision = 0.15; //Comision del 15%

    public EmpleadoxComision(String Name, int ID, int Horas, double Salario,
double Ventas) {
        super(Name, ID, Horas, Salario, Ventas);
    }

    public void name(String Name){
        this.Name = Name;
    }

    public void id(int ID){
        this.ID = ID;
    }

    public void horas(int Horas){
        this.Horas = Horas;
    }

    public void salario(double Salario){
        this.Salario = Salario;
    }

    public void ventas(double Ventas){
        this.Ventas = Ventas;
    }

    @Override
    protected void ingresaNombre(Empleados[] P0, int N) {
        System.out.print("\nNombre: ");
        name(sc.nextLine().toUpperCase());
    }

    @Override
    protected void ingresaID(Empleados[] P0, int N) {
        System.out.print("\nID: ");
        id(sc.nextInt());
    }

    @Override
```

```
protected void ingresaHoras(Empleados[] P0, int N) {
    System.out.print("\nHoras trabajadas: ");
    horas(sc.nextInt());
}

@Override
protected void ingresaVentas(Empleados[] P0, int N) {
    System.out.print("\nIngresa la cantidad en $ total de vetas: ");
    ventas(sc.nextDouble());
}

@Override
protected void ingresaSalario(Empleados[] P0, int N) {
    double sFinal;
    sFinal = comision * P0[N].Ventas;
    System.out.println("\nComision por venta del "+ comision+"%");
    System.out.println("Nomina basada en ventas: " + sFinal+"\n");
    salario(sFinal);
    sc.nextLine();
}

}
```

FijoxHora.java

```
import java.util.*;

public class FijoxHora extends Empleados {

    public static final double S_Base = 1255.25;
    public static final double comision = 0.15;
    static Scanner sc = new Scanner(System.in);

    public FijoxHora(String Name, int ID, int Horas, double Salario, double
Ventas) {
        super(Name, ID, Horas, Salario, Ventas);
    }

    public void name(String Name){
        this.Name = Name;
    }

    public void id(int ID){
        this.ID = ID;
    }

    public void horas(int Horas){
        this.Horas = Horas;
    }

    public void salario(double Salario){
        this.Salario = Salario;
    }

    public void ventas(double Ventas){
        this.Ventas = Ventas;
    }

    @Override
    protected void ingresaNombre(Empleados[] P0, int N) {
        System.out.print("\nNombre: ");
        name(sc.nextLine().toUpperCase());
    }

    @Override
    protected void ingresaID(Empleados[] P0, int N) {
        System.out.print("\nID: ");
        id(sc.nextInt());
    }
}
```

```

@Override
protected void ingresaHoras(Empleados[] P0, int N) {
    System.out.print("\nHoras trabajadas: ");
    horas(sc.nextInt());
}

@Override
protected void ingresaVentas(Empleados[] P0, int N) {
    System.out.print("\nIngresa la cantidad en $ total de vetas: ");
    ventas(sc.nextDouble());
}

@Override
protected void ingresaSalario(Empleados[] P0, int N) {
    double sFinal;
    sFinal = P0[N].Ventas *comision;
    System.out.println("Salario Base: "+S_Base + " + " + "Comisiones: "+
sFinal+"\n");
    sFinal += S_Base;
    System.out.println("Nomina: "+sFinal+"\n");
    salario(sFinal);
}
}

```

Personal.java

```
public class Personal {

    public static void mostrar(Empleados [] P0,int N){
        for(int i=0; i<N; i++){
            System.out.println("\tEmpleado");
            System.out.println("Nombre: "+P0[i].Name);
            System.out.println("ID: "+P0[i].ID);
            System.out.println("Horas: "+P0[i].Horas);
            if(P0[i].Ventas != 0)
                System.out.println("Ventas: "+P0[i].Ventas);
            System.out.println("Nomina: "+P0[i].Salario + "\n");
        }
    }

    public static void Empleado(Empleados [] P0, int N){
        P0[N].ingresaNombre(P0, N);
        P0[N].ingresaID(P0, N);
        P0[N].ingresaHoras(P0, N);
        P0[N].ingresaSalario(P0, N);
    }

    public static void Vendedor(Empleados [] P0, int N){
        P0[N].ingresaNombre(P0, N);
        P0[N].ingresaID(P0, N);
        P0[N].ingresaHoras(P0, N);
        P0[N].ingresaVentas(P0, N);
        P0[N].ingresaSalario(P0, N);
    }

}
```