

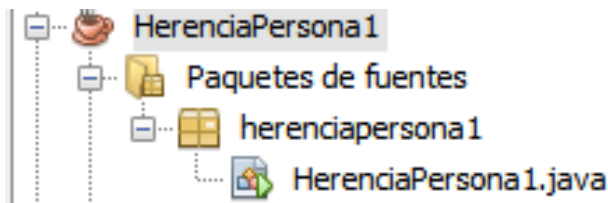
Guía 1 Herencia

Asignatura Programación II en Java

Docente: José Souza Ordenes

Objetivo, aplicar:

- Clases, métodos, herencia



```
package herenciapersona1;
```

```
// Clase Empleado
```

```
class Empleado{  
    String nombre;  
    String cedula;  
    int edad;  
    boolean casado;  
    double salario;
```

```
    Empleado(){  
    }
```

```
    Empleado(String nomb, String cedula, int edad, boolean casado, double salario) {
```

```
nombre = nomb; // si el parametro tiene un nombre diferente al atributo de la clase
this.cedula = cedula;
this.edad = edad;
this.casado = casado;
this.salario = salario;
```

```
}
```

```
public String Clasificacion(){
    String c="";
    if (edad <= 21){
        c = "Principiante";
    }
    if ((edad >=22) & (edad <= 35)){
        c = "Intermedio";
    }
    if (edad > 35){
        c = "Senior";
    }
    return c;
}
```

```
public void ImprimirEmpleado(){
    System.out.println ("\n\n Antecedente Empleado .clase padre. ");
    System.out.println("    Nombre :"+nombre);
    System.out.println("    Cedula :"+cedula);
    System.out.println("    Edad  :"+edad);
    if (casado) {
        System.out.println("    Casado");
    }
    else
    {
        System.out.println("    Soltero");
    }
    System.out.println("    Salario :"+salario);
}
```

```
public void AumentarSalario(double incremento){
    salario = salario * (1 + incremento);
}
```

```
} // Fin de la Clase Empleado
```

// sub clase Programador

```
class Programador extends Empleado{
// atributos de la sub-clase
    int lineasDeCodigoPorHora;
    String lenguajeDominante;

// constructor de la sub-clase
Programador(String nomb, String cedula, int edad, boolean casado, double salario,
int lineasCodigoHora, String lenguajeDominante){
    // super □ llamada a la clase padre
    nombre = nomb; // si el parametro es diferente, el atributo de la clase super
se asume
    super.cedula = cedula;
    super.edad = edad;
    super.casado = casado;
    super.salario = salario;
    // this □ atributos de esta clase
    lineasDeCodigoPorHora = lineasCodigoHora; // atributo de la subclase, si es
diferente al parametro, no requiere this
    this.lenguajeDominante = lenguajeDominante;
}
```

// constructor nulo

```
Programador(){
}
public void ImprimirProgramador(){
System.out.println("\n Antecedentes del Programador :");
System.out.println(" Nombre      :"+ nombre);
System.out.println(" Cedula      :"+ this.cedula);
System.out.println(" Edad       :"+ this.edad);
System.out.println(" Lineas de Codigo :"+this.lineasDeCodigoPorHora);
System.out.println(" Lenguaje Dominante :"+lenguajeDominante);
}
```

}// Fin subclase Programador

// Programa principal

```
public class HerenciaPersona1 {
    // static String cargo;
    public static void main(String[] args){
        Empleado e = new Empleado("Orlando Mena","12345",17,true,2000);
        e.ImprimirEmpleado();
        String cargo= e.Clasificacion();
        System.out.println (" El cargo es  del empleado es :"+cargo);

        Programador p1 = new
        Programador("Flavio","3456",20,false,100000,123,"Visual Basic");

        p1.nombre = "Carmen ";
        p1.salario = 250000;
        //
        p1.AumentarSalario(0.25);
        System.out.print ("\n\n El objeto p1 hace uso de la clase padre -->
p1.ImprimirEmpleado");

        p1.ImprimirEmpleado();
        System.out.print ("\n\n Imprimir objeto p1 ..subclase Programador.-->
p1.ImprimirProgramador");

        p1.ImprimirProgramador();
    }
}
```

Salida del Programa:

```
Salida - HerenciaPersona1 (run) X
run:
Antecedente Empleado .clase padre.
Nombre :Orlando Mena
Cedula :12345
Edad :17
Casado
Salario :2000.0
El cargo es del empleado es :Principiante

El objeto p1 hace uso de la clase padre --> p1.ImprimirEmpleado

Antecedente Empleado .clase padre.
Nombre :Carmen
Cedula :3456
Edad :20
Soltero
Salario :312500.0

Imprimir objeto p1 ..subclase Programador..--> p1.ImprimirProgramador
Antecedentes del Programador :
Nombre :Carmen
Cedula :3456
Edad :20
Lineas deCodigo :123
Lenguaje Dominante :Visual Basic
BUILD SUCCESSFUL (total time: 0 seconds)
```