

UNIDAD 9

PROGRAMACIÓN ORIENTADA A OBJETOS II

EVALUABLE 5. PLANTILLA DE EMPLEADOS

PROGRAMACIÓN
CFGS DAW

Paco Aldarias
paco.aldarias@ceedcv.es
2020/2021
09/02/21 10:10:11

REVISIÓN

4/2/2021. Añadir una captura de la ejecución donde aparezca vuestro nombre en el primer objeto creado.

9/2/2021. Página 2. Rectificado donde pone profesor debe poner empleado.

SUMARIO

1. INTRODUCCIÓN.....	2
2. DESCRIPCIÓN DEL PROBLEMA.....	2
3. PRINCIPIO DE ENCAPSULACIÓN.....	3
4. IDENTIFICAR LA PRÁCTICA DEBIDAMENTE.....	3
5. FUNCIÓN MAIN.....	3
6. CLASE PLANTILLA.....	3
7. EJECUCIÓN.....	4
8. DIAGRAMA DE CLASES.....	4
9. CALIFICACIÓN.....	5
10. EJECUCIÓN.....	5
11. ENTREGA.....	6

EVALUABLE 5. PLANTILLA DE EMPLEADOS

1. INTRODUCCIÓN

En la siguiente evaluables vamos a practicar ArrayList, Herencia de clases, Clases Abstractas y Funciones abstractas.

2. DESCRIPCIÓN DEL PROBLEMA

Se desea crear un lista de empleados para posteriormente calcular la nómina total. La plantilla de empleados será un ArrayList.

Tendremos los datos de **Persona** con el nombre, apellidos y la edad.

Tendremos datos de **Empleado** que es su id de empleado, así como los que herede de Persona. Un empleado debe tener un método importeNomina que deberá ser empleado por la clase que la herede.

Un empleado puede ser de dos tipos que son el empleado **Fijo** y el empleado **Contratado**. Tanto el fijo como el contratado, heredan todos los atributos de empleado e implementan los método importeNomina(). El empleado fijo el importe de la nómina será de 1500. El contratado el importe de la nómina será de 1000.

Del empleado **Contratado** nos interesa saber los días de contrato.

Se deberá mostrar los datos de los empleados existentes así como el importe total de los sueldos a pagar.

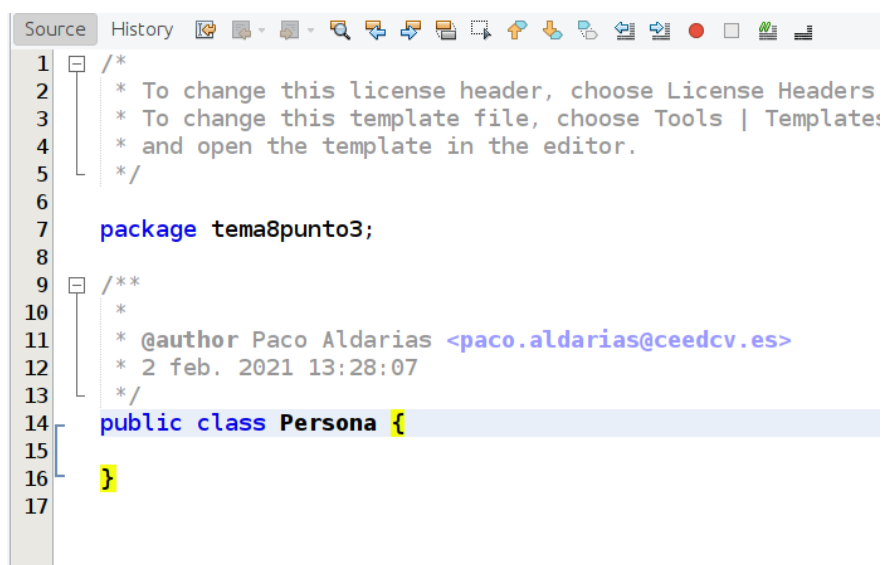
3. PRINCIPIO DE ENCAPSULACIÓN

Se deberá cumplir el principio de encapsulación en donde los atributos son privados y las funciones set y get publicas. Si ves necesario usa el atributo protected. No usar atributos public.

4. IDENTIFICAR LA PRÁCTICA DEBIDAMENTE

Todas los ficheros al inicio deberá aparecer el nombre del alumno, la fecha y el email entre comentarios. Se recomienda configurar las plantillas de Netbeans

Ejemplo:



```
1  /*
2   * To change this license header, choose License Headers
3   * To change this template file, choose Tools | Template
4   * and open the template in the editor.
5   */
6
7  package tema8punto3;
8
9  /**
10   *
11   * @author Paco Aldarias <paco.aldarias@ceedcv.es>
12   * 2 feb. 2021 13:28:07
13   */
14  public class Persona {
15
16  }
17
```

5. FUNCIÓN MAIN

La función main debe crear objetos de empleados contratados y fijos. Los objetos contratados y fijos deben ser añadidos a una clase que disponga de un arraylist donde almacenar los empleados que se podría llama Plantilla. El objeto plantilla dispondrá de una función que imprima los datos de la plantilla, la cual es llamada desde el main. Seguidamente el main mostrará el importe total de la nómina de todos los empleados.

6. CLASE PLANTILLA

Esta clase hace de almacén de objetos de la clase empleados usando una

arraylist. Los empleado pueden fijos o contratados.

Funciones recomendadas a implementar:

public void addEmpleado(Empleado empleado)

public void imprimirPlantilla()

public float importeTotalNominaEmpleados()

La función imprimirPlantilla deberá recorrer el arraylist. Se recomienda usar la instrucción **instanceof** para diferencia los objetos de la clase fijo de contratado.

7. EJECUCIÓN

```
Deleting: /home/paco/NetBeansProjects/pacoaldariasprg2021eva4/build/built-jar.properties
deps-jar:
Updating property file: /home/paco/NetBeansProjects/pacoaldariasprg2021eva4/build/built-jar.properties
Compiling 1 source file to /home/paco/NetBeansProjects/pacoaldariasprg2021eva4/build/classes
compile-single:
run-single:
Se procede a mostrar los datos de los empleados existentes en la plantilla
1. Nombre: Pepe Apellidos: Reina Edad: 45 IdEmpleado: 1258978-J Dias contrato: 12
Tipo de este empleado:Contratado Nómina de este empleado: 1000.0
2. Nombre: Pedro Apellidos: Trujillo Edad: 87 IdEmpleado: 5666899-L Dias contrato: 30
Tipo de este empleado:Contratado Nómina de este empleado: 1000.0
3. Nombre: Rosa Apellidos: Arias Edad: 76 IdEmpleado: 89788993-X Dias contrato: 50
Tipo de este empleado:Contratado Nómina de este empleado: 1000.0
4. Nombre: Andres Apellidos: Iniesta Edad: 23 IdEmpleado: 78899553-H
Tipo de este empleado:Fijo Nómina de este empleado: 1500.0
5. Nombre: Juan Apellidos: Espinosa Edad: 49 IdEmpleado: 8878994-L
Tipo de este empleado:Fijo Nómina de este empleado: 1500.0
6. Nombre: Luis Apellidos: Davila Edad: 49 IdEmpleado: 85556661-F
Tipo de este empleado:Fijo Nómina de este empleado: 1500.0
El importe de las nóminas de los empleados que consta en la plantilla es 7500.0 euros
BUILD SUCCESSFUL (total time: 0 seconds)
```

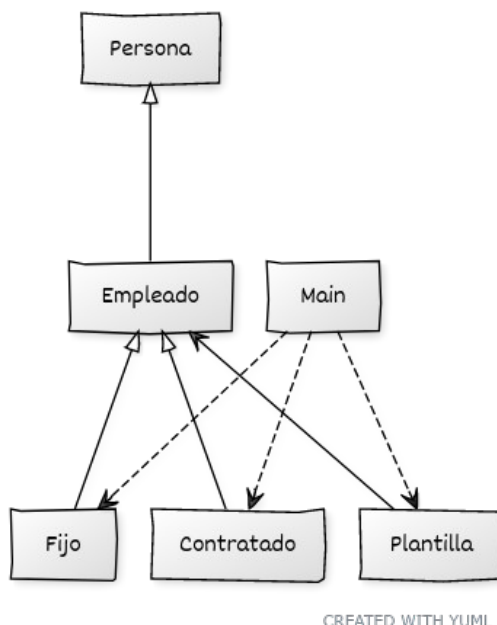
Rellenar el primer nombre que pone Pepe Reina, con vuestro nombre y hacer una captura, para subirla junto con el código.

8. DIAGRAMA DE CLASES

Creado con: <https://yuml.me/>

Script:

```
[Persona]^-[Empleado]
[Empleado]^-[Fijo]
[Empleado]^-[Contratado]
[Empleado]<-[Plantilla]
[Main]-.->[Plantilla]
[Main]-.->[Fijo]
[Main]-.->[Contratado]
```



9. CALIFICACIÓN

1. Clase Persona. Tiene atributos nombre, apellidos y edad. 1 punto.
2. Clase Empleado. Es abstracta y hereda de Persona. Tiene atributo idempleado. Tiene la función abstracta importeNomina(). 1 punto.
3. Clase Fijo. Hereda de Empleado. Implementa importeNomina() 2 punto.
4. Clase Contratado. Hereda de Empleado. Tiene atributo dias de contrato. Implementa importeNomina() 2 puntos.
5. Clase Plantilla. Crea una arraylist para las funciones addEmpleado(), imprimirplantilla(), importetotalnominaempleados(). 2 punto.
6. Clase Main. Crea objetos Fijo y Contratado para añadirlos a la plantilla y posteriormente listarlos y poder calcular la suma de las nóminas. 2 Punto.

10. ENTREGA

Entregar la url del repositorio creado en la evaluable anterior. Crear las clases en una carpeta llamada evaluable5. Añadir una captura de la ejecución donde aparezca vuestro nombre en el primer objeto creado.



Reconocimiento - NoComercial - CompartirIgual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.