



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesora:*

Rocío Alejandra Aldeco Pérez

*Asignatura:*

Programación Orientada a Objetos

*Grupo:*

6

*No de Práctica(s):*

6

*Integrante(s):*

Ugalde Velasco Armando

*No. de Equipo de  
cómputo empleado:*

*No. de Lista o Brigada:*

*Semestre:*

2021-1

*Fecha de entrega:*

13 de noviembre de 2020

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_



## Práctica de Estudio 6: Organización de clases

Programación Orientada a Objetos Grupo 6

Facultad de Ingeniería

Departamento de Computación

**Objetivo de la práctica:** Organizar adecuadamente las clases según su funcionalidad o propósito bajo un **namespace** o **paquete**.

Realiza las siguientes actividades después de leer y revisar en clase la **Práctica de Estudio 6: Organización de Clases**.

1. Usando como base el programa que creaste en la **Práctica 4**, realiza las siguientes actividades para entender el uso y creación de paquetes:
  - a) Todas las clases que generaste (excepto la clase principal) ponlas en un archivo nuevo llamado "**practica6.java**". Ese archivo deberá volverse un paquete. Recuerda poner la ruta correcta y usar la palabra reservada **package**.
  - b) Genera un archivo "**Main.java**" en donde incluirás la clase principal exclusivamente. Como esta clase hace uso del paquete que generas en el punto anterior deberás hacer uso de la palabra reservada **import**.
  - c) Ahora deberás compilar ambos archivos **.java**. Recuerda tener cuidado con las rutas donde cada uno se encuentra. Prueba su correcto funcionamiento.
  - d) Finalmente, genera el archivo **.jar** de este programa. Recuerda que como estas usando un paquete propio deberás incluir **practica6.class** y **Main.class**.
  - e) Cuando estés seguro de que tu programa es correcto, súbelo a **Alphagrader**. Verás que los casos de prueba son los mismos que en la practica 4, por lo que no deberías de tener algún problema.
  
2. Usando como base el programa que creaste en la **Práctica 4**, realiza las siguientes actividades para entender el funcionamiento de **Javadoc**:
  - a) Documenta tu paquete siguiendo el formato establecido por **Javadoc** [1] Debe de contener lo siguiente:
    - Información del autor
    - Versión
    - Una descripción general
    - La descripción general de cada clase
    - La explicación de los constructores y métodos de cada clase
  - b) Una vez que este todo documentado genera el archivo **html** correspondiente.
  - c) Finalmente sube tus archivos **.java** y el archivo **.html** a Schoology
  - d) Genera un documento **pdf** con la correspondiente carátula que explique este proceso y súbelo a **Schoology**.



## Práctica de Estudio 6: Organización de clases

Programación Orientada a Objetos Grupo 6

Facultad de Ingeniería

Departamento de Computación

Se agregaron los comentarios pertinentes para cada clase y miembros de éstas, utilizando los tags correspondientes en cada caso. Se escribieron breves descripciones, tratando de que éstas fueran escritas de forma concisa y aportaran más información que la presente en la firma del método. Además, se siguieron algunas recomendaciones presentes en la página de *Oracle*.

A continuación, se muestran algunos ejemplos de lo anterior:

```
* Representa un empleado con pago por hora
*
* @author Armando Ugalde Velasco
* @version 1.0
*/
public class HourlyEmployee extends Employee
{
```

### Comentario en definición de clase

```
/**
 * Porcentaje de salario a pagarle al empleado después de impuestos, dividido entre 100.
 */
private final static double NORMAL_PERCENTAGE = 0.9;
/**
 * Número de horas trabajadas por este empleado
 */
private final int workedHours;
/**
 * Pago por hora de este empleado
 */
private double hourFee;
```

### Comentarios en atributos



## Práctica de Estudio 6: Organización de clases

Programación Orientada a Objetos Grupo 6  
Facultad de Ingeniería  
Departamento de Computación

```
/**
 * Aumenta el salario de este empleado por el porcentaje proporcionado
 *
 * @param percentage El porcentaje por el que se aumentará el salario
 */
@Override
public void increaseSalaryBy(double percentage) { this.hourFee *= 1 + (percentage / 100); }

/**
 * Obtiene el número de horas trabajadas por este empleado
 * @return El número de horas trabajadas por este empleado
 */
public int getWorkedHours() { return workedHours; }
```

### Comentarios en definiciones de métodos

Además, se creó el archivo **package-info.java**, para añadir la información correspondiente a todo el paquete. A continuación, se muestra una captura de pantalla de lo anterior:

```
/**
 * Clases utilizadas en la realización de la práctica 6.
 * Representan la nómina de los empleados y a éstos últimos,
 * dependiendo de sus características.
 *
 * @author Armando Ugalde Velasco
 * @version 1.0
 */
package practica6;
```

### Comentario sobre el paquete practica6

## Referencias

- **Javadoc** de Oracle, <https://docs.oracle.com/javase/1.5.0/docs/tooldocs/windows/javadoc.html>.



## Práctica de Estudio 6: Organización de clases

Programación Orientada a Objetos Grupo 6

Facultad de Ingeniería

Departamento de Computación

### CONCLUSIONES

El lenguaje Java cuenta con una serie de convenciones establecidas que facilitan la modularización de nuestros programas. Además de la inherente presencia del paradigma orientado a objetos en el lenguaje, también es posible agrupar las clases acorde a su dominio y relaciones mediante la utilización de **paquetes**. Éstos, sin duda, representan una abstracción o un nivel de modularidad mayor al de las clases. Lo anterior tiene una gran utilidad en el desarrollo de sistemas complejos, donde es necesario modelar la información y funcionalidad de forma eficiente y clara.

Además, otro aspecto que aporta a la robustez del lenguaje es, sin duda alguna, la herramienta **Javadoc**. Ésta, nos permite generar documentación automática de nuestros programas, a partir de la realización de comentarios en las partes relevantes del código fuente.