	Carátula para entrega de prácticas
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Dra. Rocío Alejandra Aldeco Pérez

Asignatura: Programación orientada a objetos -1323

Grupo: 6

No de Práctica(s): 6

Integrante(s): Mendoza Hernández Carlos Emiliano

No. de Equipo de cómputo empleado:

No. de Lista o Brigada:

Semestre: 2023-1

Fecha de entrega: 24 de octubre del 2022

Observaciones:

CALIFICACIÓN: _____

Práctica 6.

Organización de clases.

OBJETIVO

- Organizar adecuadamente las clases según su funcionalidad o propósito bajo un *namespace* o paquete.

ACTIVIDADES

- Agrupar clases en paquetes.
- Importar clases en diversos paquetes.

INTRODUCCIÓN

Las clases de las bibliotecas estándar del lenguaje están organizadas en jerarquías de paquetes. Esta organización en jerarquías ayuda a que las personas encuentren clases particulares que requieren utilizar.

Esta bien que varias clases tengan el mismo nombre si están en paquetes distintos, Así, encapsular grupos pequeños de clases en paquetes individuales permite reusar el nombre de una clase dada en diversos contextos.

INSTRUCCIONES

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 5*, 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 usar la ruta correcta y 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 incluir 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 estes seguro de que tu programa es correcto, súbelo a Alphagrader. Verás que los casos de prueba en esta ocasión incluyen la verificación de la multiplicación y el uso de una estructura de datos.
    f) Usando como base el programa que creaste en la *Práctica 5*, realiza las siguientes actividades para entender el funcionamiento de *Javadoc*:
    g) Documenta tu paquete siguiendo el formato establecido por *Javadoc*. Debe contener lo siguiente:
        - La descripción general de cada clase.
        - La explicación de atributos, constructores y métodos de cada clase.
        - Los métodos deberán incluir su descripción además de los parámetros que solicitan y regresan (si aplica).

h) Una vez que este todo documentado genera el archivo *html* correspondiente.

2) Documenta el proceso que realizaste en la práctica y recuerda incluir conclusiones.

DESARROLLO

Descripción de actividades realizadas

Se realizó lo siguiente en una copia del programa de la *Práctica 5.*

1) En el código anterior, el programa está conformado por 2 clases (*Polynomial* y la clase *Main*). Por lo tanto, se copió la clase *Polynomial* en una nueva clase llamada *Practica6* (los nombres de las clases se escriben, por convención, con mayúscula inicial). Sin embargo, esto tiene consecuencias a nivel de compatibilidad de código (los constructores deben tener el nombre de la clase y el tipo de dato abstracto ya no es *Polynomial* sino *Practica6*), por lo que se renombraron algunos aspectos del código para solucionar los errores. ***Nota*:** En mi opinión, este cambio no parece ser muy adecuado porque, siguiendo el patrón de modelado donde cada archivo debe tener el mismo nombre de la clase que contiene, el nombre de "*Practica6*" es menos descriptivo que "*Polynomial*".

2) Para que la clase *Practica6* forme parte del paquete *mx.unam.fi.poo* se agregó, en la primera línea del código, la sentencia

```
package mx.unam.fi.poo;
```

3) En el código de la clase *Main*, antes de la declaración de la clase, se agregó la sentencia para importar la clase *Practica6*.

```
import mx.unam.fi.poo.Practica6;
```

4) En una terminal de *PowerShell* se compilaron los archivos *.java* con los comandos:

```
javac -d . mx/unam/fi/poo/Practica6.java
javac -d . Main.java
```

5) Se generó el archivo *jar* desde la terminal de *PowerShell* con el siguiente comando:

```
jar -cvfe program.jar Main Main.class mx/unam/fi/poo/Practica6.class
```

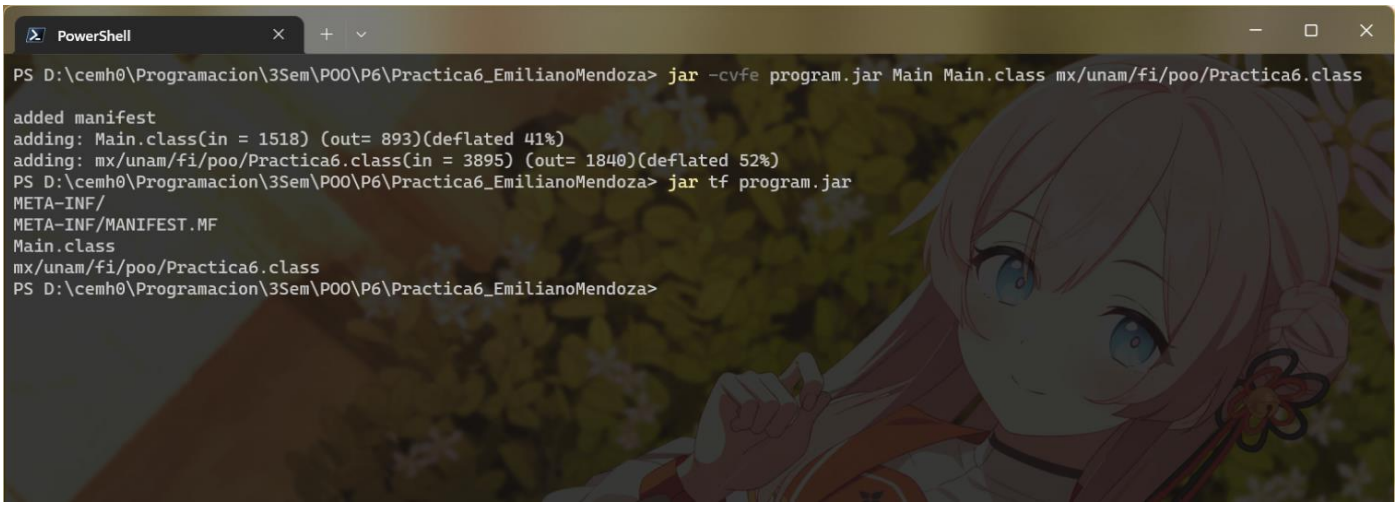
Esto incluye dentro del *jar* a la clase *Main* y a la clase *Practica6* del paquete *mx.unam.fi.poo*. Además, se indicó que la clase principal del programa es la clase *Main*.

6) Para verificar el contenido del *jar* se utilizó el comando:

```
jar tf program.jar
```

Dentro del *jar* se observa que se encuentra la jerarquía de directorios correspondientes al paquete *mx.unam.fi.poo* con el archivo compilado *Practica6.class* y adicionalmente el archivo *MANIFEST.MF* dentro de un directorio *META-INF*.

Contenido de *program.jar*



7) Para probar el correcto funcionamiento del programa, se creó un archivo de texto plano con uno de los *inputs* de Alphagrader.

Archivo *input*

```
input
1   3
2   6 7 3 -2
3   4
4   5 -3 0 2 1
5   -2
6   3
```

8) Se ejecuta el programa desde una terminal de símbolo del sistema con el comando:

```
java -classpath program.jar Main < input
```

Salida del programa

```
D:\cemh0\Programacion\3Sem\POO\P6\Practica6_EmilianoMendoza>java -classpath program.jar Main < input
6x^3+7x^2+3x-2
5x^4-3x^3+2x+1
5x^4+3x^3+7x^2+5x-1
-5x^4+9x^3+7x^2+x-3
18x^3+21x^2+9x-6
15x^4-9x^3+6x+3
73
-129

D:\cemh0\Programacion\3Sem\POO\P6\Practica6_EmilianoMendoza>
```

9) Una vez que se comprobó su funcionamiento, se creó el *zip* con el archivo *jar* y se subió en Alphagrader.

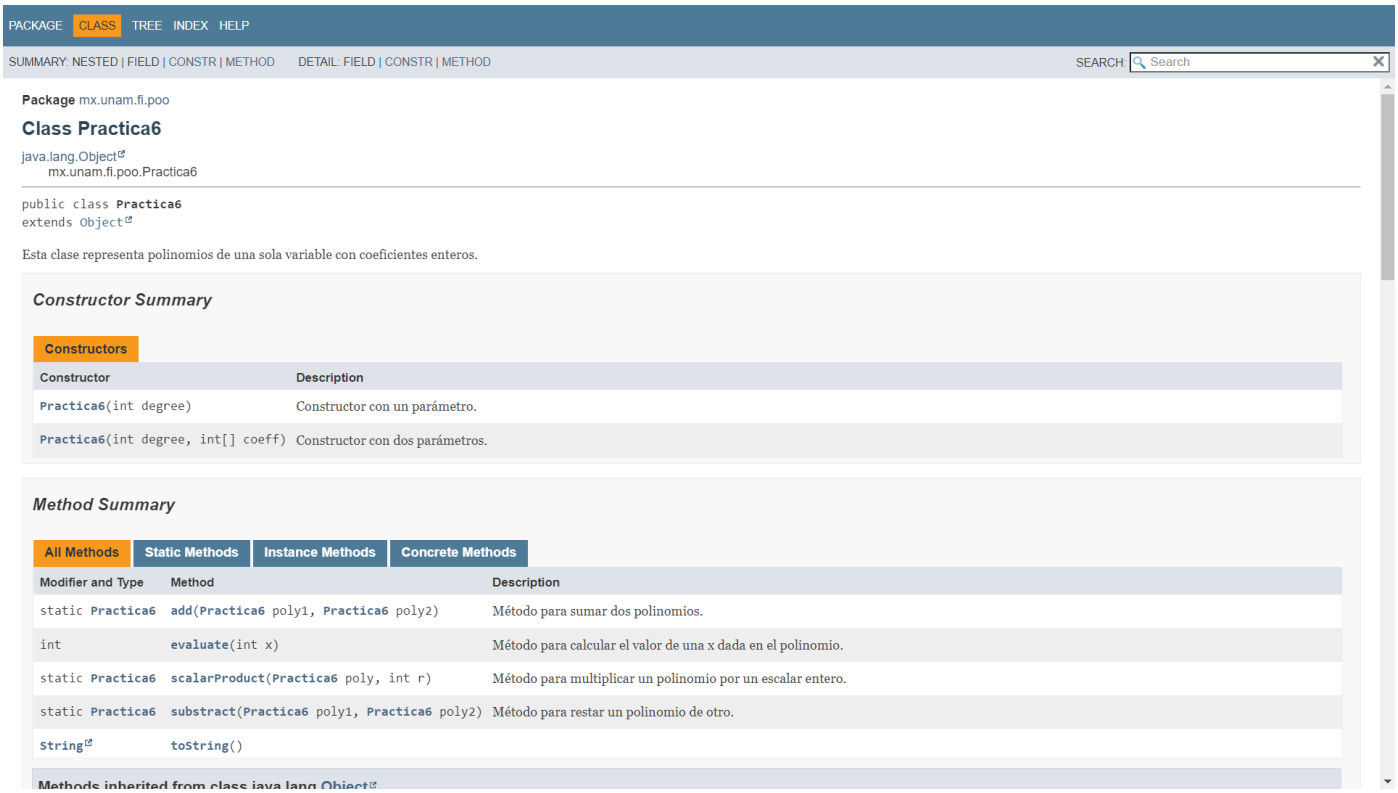
10) Después de que el programa pasó todas las pruebas, se agregaron comentarios de documentación al código de *Practica6*. Se describió la clase principal, así como sus constructores, métodos y atributos.

11) Para crear el *javadoc* se ejecutó el siguiente comando en la terminal:

```
javadoc -d ../Doc mx.fi.unam.poo
```

12) Se revisó la documentación creada en la carpeta *Doc*. Una vez revisado que todo funciona correctamente, concluye el desarrollo de la práctica.

Documentación en formato *javadoc* de la clase *Practica6*



PACKAGE CLASS TREE INDEX HELP

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD SEARCH: Search

Package mx.unam.fi.poo

Class Practica6

java.lang.Object
mx.unam.fi.poo.Practica6

```
public class Practica6
extends Object
```

Esta clase representa polinomios de una sola variable con coeficientes enteros.

Constructor Summary

Constructors

Constructor	Description
Practica6(int degree)	Constructor con un parámetro.
Practica6(int degree, int[] coeff)	Constructor con dos parámetros.

Method Summary

All Methods | Static Methods | Instance Methods | Concrete Methods

Modifier and Type	Method	Description
static Practica6	add(Practica6 poly1, Practica6 poly2)	Método para sumar dos polinomios.
int	evaluate(int x)	Método para calcular el valor de una x dada en el polinomio.
static Practica6	scalarProduct(Practica6 poly, int r)	Método para multiplicar un polinomio por un escalar entero.
static Practica6	substract(Practica6 poly1, Practica6 poly2)	Método para restar un polinomio de otro.
String	toString()	

Methods inherited from class java.lang.Object

CONCLUSIONES

En esta práctica se organizó la estructura del programa desarrollado en la práctica anterior, siguiendo una jerarquía de paquetes. Para ello, se declaró un paquete que se importó desde la clase principal. Este procedimiento ilustra de manera sencilla el funcionamiento de los paquetes y sus ventajas (agrupación de clases, mejor control de la accesibilidad de clases).

Además, se creó un archivo *jar* con las clases compiladas del programa con el fin de facilitar el traslado de la aplicación. Se aprendió que se pueden ejecutar clases dentro de un archivo *jar* desde la terminal.

Por último, se creó una documentación del programa siguiendo el formato *javadoc*, describiendo la clase principal, así como sus métodos y atributos. Es muy importante conocer esta herramienta puesto que maneja un estándar universal que hace que sea más fácil de comprender el código para otras personas.

Dado que se organizaron adecuadamente las clases en un paquete, se concluye que se cumplieron los objetivos planteados.

REFERENCIAS

- *Barnes David, Kölling Michael*
  Programación Orientada a Objetos con Java.
  Tercera Edición.
  Madrid
  Pearson Educación, 2007

- *Deitel Paul, Deitel Harvey*
  Cómo programar en Java.
  Séptima Edición.
  México
  Pearson Educación, 2008

- *Martín, Antonio*
  Programador Certificado Java 2.
  Segunda Edición.
  México
  Alfaomega Grupo Editor, 2008

- *Dean John, Dean Raymond*
  Introducción a la programación con Java.
  Primera Edición.
  México
  Mc Graw Hill, 2009

Yo, Carlos Emiliano Mendoza Hernández, hago mención que esta práctica fue de mi autoría.