



Instituto Politécnico Nacional
Unidad Profesional
Interdisciplinaria De Ingeniería
Campus Zacatecas



Ingeniería en Sistemas Computacionales

Práctica 1:

¿Cómo compilar programas orientados a objetos?

Alumna: Vanessa Melenciano Llamas

Boleta: 2020670081

Profesora: Monreal Mendoza Sandra Mireya

Materia: Programación Orientada a Objetos

Grupo: 2CM2

Fecha: 3 de octubre de 2020

Índice

Introducción	3
Objetivos	4
Desarrollo	4
Código fuente	4
Funcionamiento	6
Errores detectados	9
Posibles mejoras	9
Preguntas	9
Conclusiones	11
Bibliografía	11

Introducción

La programación orientada a objetos es un paradigma de programación en el que los programas se ven como formados por entidades llamadas objetos que recuerdan su propio estado interno y que se comunican entre sí mediante el paso de mensajes que se intercambian con la finalidad de cambiar sus estados internos, compartir información y solicitar a otros objetos el procesamiento de dicha información. (Pérez, 2020), es decir que los Lenguajes Basados en Objetos poseen una sintaxis y una semántica que permiten la creación de objetos.

Para la comprensión correcta de la POO es necesario saber varios términos como: *métodos* que son las funciones definidas dentro de una clase, *un objeto* que es una colección de operaciones que comparten un estado (todas referidas al mismo), encapsulación de un conjunto de operaciones y métodos que pueden ser invocados externamente y que recuerdan el estado; y, por último, *los campos* son los datos que va a almacenar la clase o los objetos de esa clase.

Siempre se dice que en POO se busca tener una determinada serie de objetos que se comunican mediante mensajes; un mensaje se interpreta como una llamada a uno de los procedimientos del objeto pues, como hemos comentado, sus métodos determinan su comportamiento, son su “interfaz” para interactuar con él.

Para este curso se utilizará el lenguaje Java porque “ofrece en sí una implementación muy limpia de los conceptos más importantes de la orientación a objetos y funciona bien en la enseñanza, como lenguaje introductorio. Su popularidad asegura una inmensa fuente de recursos de apoyo.” (Barnes & Kolling, 2007)

Para la correcta utilización de Java, se pueden usar varios programas disponibles, pero de los más fáciles de utilizar está JDK. Java Development Kit o JDK es un software para los desarrolladores de Java. Incluye el intérprete Java, clases Java y herramientas de desarrollo Java (JDT): compilador, depurador, desensamblador, visor de applets, generador de archivos de apéndice y generador de documentación.

Compilador es un programa que traduce de un lenguaje de alto nivel a un lenguaje de bajo, depurador es un programa usado para probar y depurar (eliminar) los errores de otros programas y desensambladores que traducen de código máquina a lenguaje ensamblador (López, Hernández, Sánchez, & Cruz, 2009)

El JDK permite escribir aplicaciones que se desarrollan una sola vez y se ejecutan en cualquier lugar de cualquier máquina virtual Java. Las aplicaciones Java desarrolladas con el JDK en un sistema se pueden usar en otro sistema sin tener que cambiar ni recompilar el código. Los archivos de clase Java son portables a cualquier máquina virtual Java estándar. (IBM Knowledge Center, s.f.)

Objetivo

El objetivo principal de esta práctica, es introducirnos a la programación orientada a objetos, comenzando a entender los conceptos básicos. Para lograrlo, es necesario la utilización del lenguaje de programación Java que se utilizará en el curso, y para comenzar con ello, es necesario conocer la interfaz de NetBeans, que es lo que se espera en esta práctica.

Desarrollo

Código Fuente

Ejercicio 1

```
* @author Vanessa
*/
public class Ejercicio1{
    public static void main(String[] args){
        System.out.println("Ejercicio1\n");
    }
}
```

Código que sólo imprime "Ejercicio1" en pantalla.

Ejercicio 2

```
/* @author Vanessa
 */
import java.util.*;
public class Ejercicio2{
    public static void main (String[] args){
        Scanner reader = new Scanner(System.in);
        int n1;
        int n2;
        int suma;

        System.out.println("Ingrese el 1er numero: ");
        n1=reader.nextInt();

        System.out.println("Ingrese el 2do numero: ");
        n2=reader.nextInt();

        suma = n1 + n2;

        System.out.println("suma de " + n1 + " y " + n2 + " es " + suma);
        System.exit(0);
    }
}
```

Este código pide dos números al usuario, después de guardarlos en una variable cada uno, realiza la suma y la muestra en pantalla.

Ejercicio 3:

```
/* @author Vanessa
 */
import java.util.*;
public class LecturaConsola{
    public static void main(String[] args){
        Scanner reader = new Scanner(System.in);
        String nombre = "";
        int edad;

        System.out.println("Ingrese su nombre: ");
        nombre=reader.nextLine();
        System.out.print("Ingrese su edad: ");
        edad=reader.nextInt();

        System.out.println("Nombre: " + nombre);
        System.out.println("Edad: "+edad);
    }
}
```

Este código pide al usuario ingresar su nombre, después su edad, guardando en una variable cada dato, al final los muestra en pantalla, señalando cual es cada uno.

Ejercicio 4

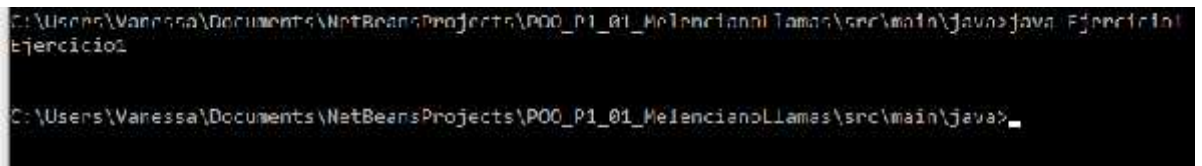


Este código pide al usuario escribir su nombre, para después mostrarlo con un mensaje, pero a diferencia de los anteriores, lo hace por medio de ventanas, no en la consola.

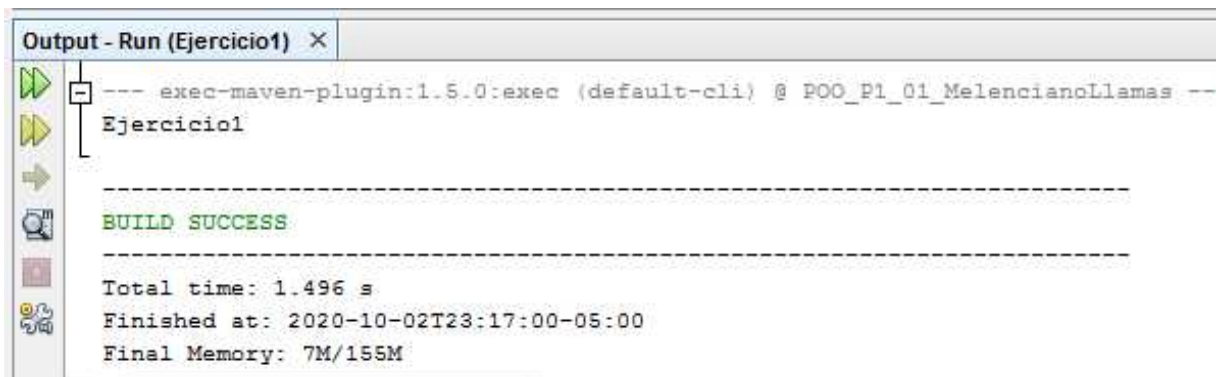
Funcionamiento

Ejercicio 1

➤ Desde consola



➤ Desde el IDE de desarrollo

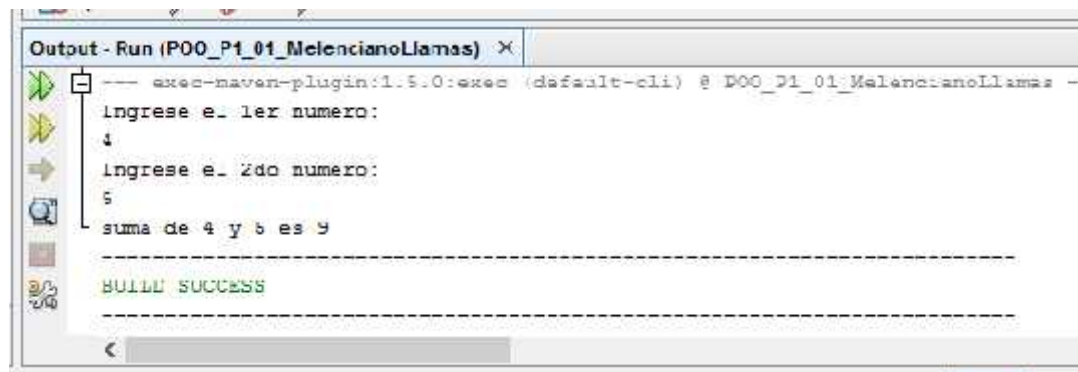


Ejercicio 2

➤ Desde consola

```
C:\Users\Vanessa\Documents\NetBeansProjects\POO_P1_01_MelencianoLlamas>src\main\java>java Ejercicio2.java
C:\Users\Vanessa\Documents\NetBeansProjects\POO_P1_01_MelencianoLlamas\src\main\java>java Ejercicio2
Ingrese el 1er numero:
4
Ingrese el 2do numero:
5
suma de 4 y 5 es 9
C:\Users\Vanessa\Documents\NetBeansProjects\POO_P1_01_MelencianoLlamas\src\main\java>
```

➤ Desde el IDE de desarrollo

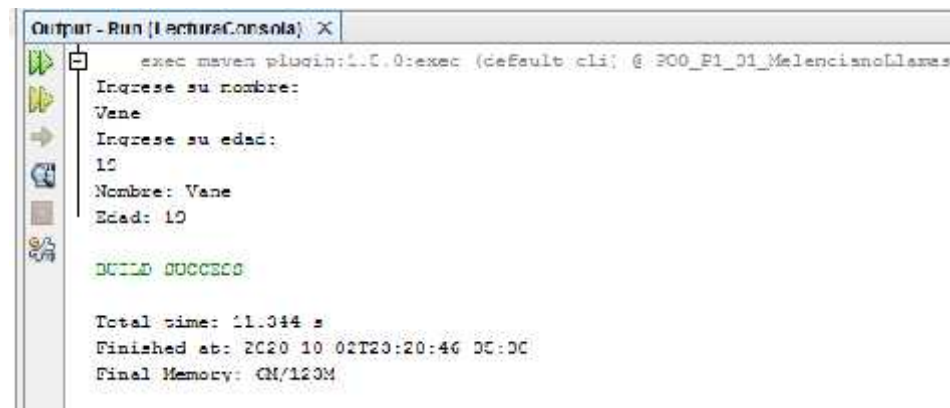


Ejercicio 3

➤ Desde consola

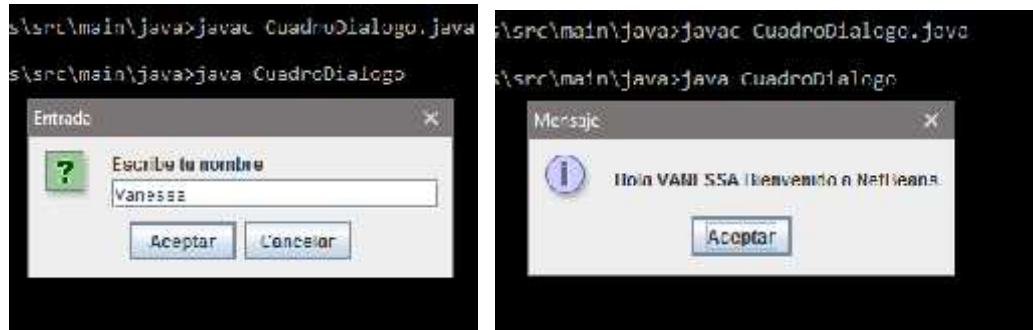
```
C:\Users\Vanessa\Documents\NetBeansProjects\POO_P1_01_MelencianoLlamas\src\main\java>java LecturaConsola.java
C:\Users\Vanessa\Documents\NetBeansProjects\POO_P1_01_MelencianoLlamas\src\main\java>java LecturaConsola
Ingrese su nombre:
Vane
Ingrese su edad:
10
Nombre: Vane
Edad: 10
C:\Users\Vanessa\Documents\NetBeansProjects\POO_P1_01_MelencianoLlamas\src\main\java>
```

➤ Desde el IDE de desarrollo



Ejercicio 4

➤ Desde consola



➤ Desde el IDE de desarrollo



Errores detectados

El primer problema que surgió en la práctica, fue la instalación de JDK, debido a una pestaña que abrió después de la instalación, que no se podía cerrar, pero después de solucionar ese problema, la instalación del NetBeans fue muy sencillo. Después hubo una complicación al no tener dominio del programa NetBeans, porque no se creaba correctamente los proyectos ni las clases, lo que causaba que no compilara y marcara errores, pero posteriormente a buscar ayuda para poder

solucionar este problema. Al pasar a tener los códigos listos para compilar, hubo pequeños errores a causa de tener mal el código por mala transcripción.

Los últimos errores detectados, fueron al usar la consola, por falta de dominio de esta, lo que causó un retraso en la práctica.

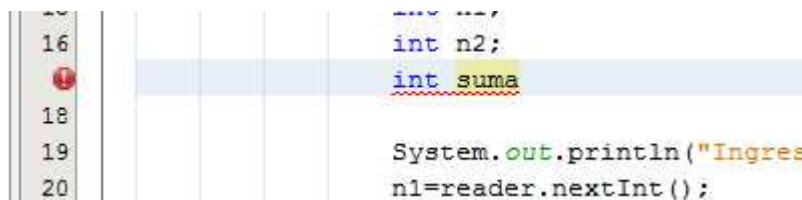
Posibles mejoras

Hay muchas cosas que se pueden mejorar, no solo para la resolución de las prácticas, sino para dominar y aprender más de tema importantes para la carrera. Entre ellos, está aprender a utilizar la consola, aprender de la sintaxis de java, y saber utilizar el programa NetBeans. De manera general, eso sería.

Preguntas

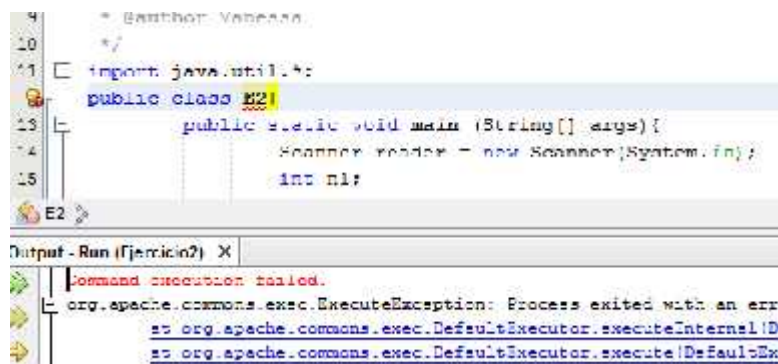
1. ¿Qué ocurre en el IDE si tenemos un error de sintaxis como la omisión del punto y coma?

No deja correr el programa y marca un error en el código.



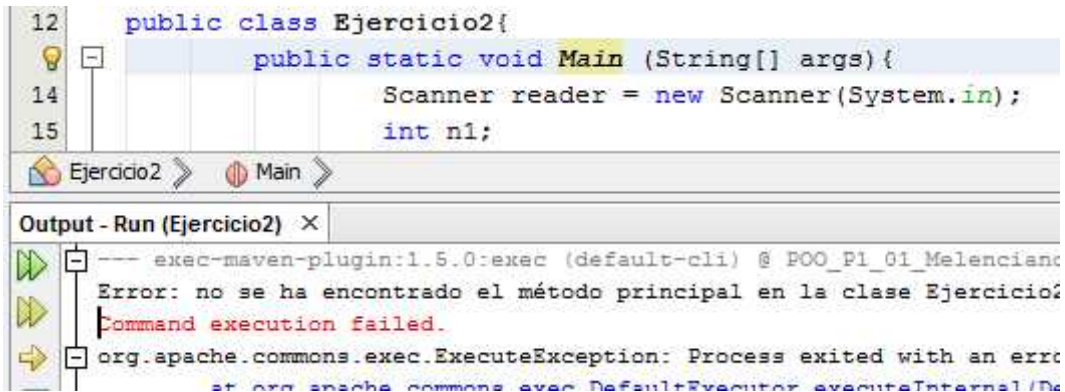
2. ¿Qué pasa si cambiamos la palabra que está enseguida de la palabra “class”? ¿es posible ejecutar?, ¿por qué?

Marca error y no deja compilar, esto sucede porque la clase está guardada con el nombre de “Ejercicio2” y al cambiar el código el nombre, no puede relacionarla.



3. ¿Qué pasa si cambiamos la palabra “main” por “Main”?

No deja seguir, marcando error.



```
12 public class Ejercicio2{
13     public static void Main (String[] args){
14         Scanner reader = new Scanner(System.in);
15         int n1;
```

Output - Run (Ejercicio2) X

```
--- exec-maven-plugin:1.5.0:exec (default-cli) @ POO_Pl_01_Melencianc
Error: no se ha encontrado el método principal en la clase Ejercicio2
Command execution failed.
org.apache.commons.exec.ExecuteException: Process exited with an error
at org.apache.commons.exec.DefaultExecutor.executeInternal(DefaultExecutor.java:404)
```

4. ¿En qué se parece el lenguaje Java al lenguaje C?

- En que todas las líneas de comando, tienen que terminar con punto y coma para evitar error.
- Es muy parecido su comando para imprimir en pantalla, en C es “printf()” y en Java “println()”.
- Ambos utilizan funciones o métodos.
- Los dos lenguajes usan clasificación en las variables (int, float...)
- Java utiliza las mismas palabras clave que C para indicar el control de acceso: private, public y protected.
- Los tipos básicos de datos son similares en java y C
- Entre más.

Conclusiones

El objetivo original de la práctica, fue cumplido, con un poco de complicaciones en el transcurso, pero se logro comprender el concepto general de la programación orientada a objetos, así como conocer lo básico del programa NetBeans y de la consola. Es importante tener todos estos conceptos claros, ya que es la base de la POO, y para logras una correcta introducción a los siguientes temas.

En está practica se pudo ver dos diferentes formas de compilar, y comprara la facilidad o dificultad de cada una, y así poderlas utilizar con más confianza, comprender la importancia de esto cuando se está aprendiendo un nuevo lenguaje, y POO.

Bibliografía

Barnes, D., & Kolling, M. (2007). *Programación orientada a objetos con Java* (3ª edición ed.). Madrid, España: PEARSON EDUCACIÓN.

IBM Knowledge Center. (s.f.). Obtenido de https://www.ibm.com/support/knowledgecenter/es/ssw_ibm_i_73/rzaha/sunjdk.htm

López, B., Hernández, B., Sánchez, M., & Cruz. (2009). PROGRAMACIÓN FÁCIL en el proceso enseñanza-aprendizaje. *Innovación Educativa*, pp. 61-71.

Pérez, R. (2020). Programación orientada a objetos. *IES Doñana*.