|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Ing. Guadalupe Lizeth Parrales Romay |
| *Asignatura:* | Programación Orientada a Objetos |
| *Grupo:* | 04 |
| *No de Práctica(s):* | 1 |
| *Integrante(s):* | Díaz Hernández Marcos Bryan |
| *No. de Equipo de cómputo empleado:* | Equipo personal. |
| *No. de Lista o Brigada:* | 14 |
| *Semestre:* | 2021-1 |
| *Fecha de entrega:* | 6 de octubre del 2020 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**Objetivo de la practica**

Identificar y probar el entorno de ejecución y el lenguaje de programación orientado a objetos a utilizar durante el curso.

**Introducción**

En la práctica y en este reporte se trabajó en función de fortalecer los conocimientos teóricos de la Programación Orientada a Objetos, con el motivo de poder aplicar los temas vistos en clase, además de lograr identificar y entender cómo funcionan los entornos de desarrollo.

En cuanto a los entornos de desarrollo se buscó el entender las clases, los objetos, las diferencias entre los argumentos del main, los modificadores de acceso, en general familiarizarnos con entorno y poder comprenderlo para empezar a trabajar en él, con la intención de poder implementar y utilizar un lenguaje de programación que en este curso será Java.

Por ultimo al implementar el lenguaje se busca el aprendizaje de la forma correcta de escribir el código y los posibles errores que se pueden cometer, incluyendo la manera más adecuada de compilación de los programas y como ejecutarlos. Es mas una muestra de los elementos básicos de los cuales se compone Java.

Breve introducción de Java.

En los orígenes de Java, no era lo que hoy es y por supuesto que tuvo que irse modificando, en 1990 se comenzó el trabajo en Java, por la competencia entre Sun Systems y Microsoft, de hecho el surgimiento de Java fue por una lucha comercial entre los sistemas operativos de las pc, y conforme a eso el Bill Joy, fue en la búsqueda de ello, pero por medio de una gran idea, que era el lograr tareas complejas con software simple, algo bastante complejo para su tiempo y actualmente sigue siendo igual.

En el proceso se buscó el aplicar la información, es decir crear PDAs, celulares, por medio de la transferencia de información en tiempo real con networks incluidas, es decir la interacción entre elementos de un sistema, para lograr esto se comenzó a trabajar en C++, pero el lenguaje volvió todo bastante complejo y ser bastante inseguro, por lo que comenzaron desde cero, y bueno saltando algunos pasajes el resultado se denomino Oak, que con el paso del tiempo sería Java.

Algo especial de Java es la Maquina Virtual, debido a que es un lenguaje compilado e interpretado, ya que las instrucciones llegan al bytecode, no directo al sistema, esto es un proceso primordial que logra diferencial a Java de los demás lenguajes porque lo convierten en un lenguaje bastante completo en aspectos de seguridad, velocidad y portabilidad. Continuando con el bytecode, Java utiliza un compilador que permite el procesamiento de las instrucciones pero en una instancia artificial, lo que lo vuelven proceso seguro en caso de que este pudiera afectar al dispositivo, por medio de la VM y de la definición del lenguaje se evitan las redundancias en el lenguaje, es decir, no se le escapa nada.

Al mismo tiempo la implementación del interpretador, el bytecode y la VM, pueden correr en cualquier sistema operativo sin que este tenga algún efecto sobre los programas que se realicen, ya que lo que se encarga de realizar las instrucciones de cada código son las entidades antes mencionadas.

Como inducción al lenguaje Java, este es un lenguaje multipropósito, al igual que C, Python etc. Igual es importante el diferenciarlos de JavaScript, donde el primero es que usare en el curso de POO y el segundo es funciona para agregar funcionalidad dinámica a las paginas web, es decir, son dos cosas distintas. Sin embargo la sintaxis es similar a la C y C++.

Por otro lado Java es un lenguaje muy bien equipo a cuanto a métodos distintos, pertenecientes a un árbol de clases, donde se incluye, cadenas, sockets, interfaz gráfica, acceso a bases de datos, y algunas más. Por otro lado se pueden utilizar editores de texto plano para la codificación pero es recomendable el usar las IDES, para el desarrollo mas avanzado de los programas y porque cuentan con mas recursos disponibles para el programar.

**Ejercicios de la practica:**

* Ejercicio 1:

Este fue un ejercicio que consistió en varias partes de elaboración, debido a que se buscaba el entender el cómo funciona el entorno de desarrollo de Java, por lo que consistió en pruebas y errores para poder reconocer lo que se puede hacer y que no se puede hacer.

* Dificultades en el código

En este caso fue guiada la elaboración de los ejercicios por lo que no hubo mucha dificultad, pero algo que me sucedió al estar elaborando o siguiendo la elaboración de los programas fue que no recordaba los comandos de sistema y me tarde en seguir el hilo de lo que estaba haciendo la profesora, pero al ver el video de la sesión pues todo fue más claro.

* Diagrama de funcionamiento:

Normalmente en mis reportes hago diagramas, que no son de flujo, porque trato de expresar de forma más simple la forma en que resolví los ejercicios y de esta manera puedo expresar un poco la manera en que veo la solución de los ejercicios, además que añado explicaciones de las partes que considero importantes:

1. Programa

public static void main(String[] b){

System.out.println("Hola Mundo");

}

public class HolaMundo{

Se crea la clase:

El método principal (main), es el que invoca a las demás clases, métodos, y debe tener los argumentos.

Ser público, estático y no devolver valores.

Cuando la clase se define pública el nombre del archivo debe corresponder al de la clase. Y la clase publica es la principal.

En los argumentos del main, deben ir un string, sin definir la cantidad y con un identificador args (u otro)

Por buenas prácticas se recomienda el utilizar mayúsculas para el inicio del nombre de la clase.

class HolaAmigos{

public static void main(String[] b){

int holaAtodos = 1;

int holaATodos = 2;

int c = holaATodos + holaAtodos;

System.out.println("C: " + c);

}

Otra forma de clase:

Cuando no se define el modificador de acceso se toma uno por default/friendly

Para los identificadores se deben considerar las notaciones UpperCamel y LowerCamel

1. Compilación y ejecución

En la consola:

P1>java HolaMundo

P1>javac HolaMundo.java

P1>javac HolaMundo.java

P1>javac HolaMundo.java

-java: ejecuta los class creados, y en caso de error lo notifica.

-javac: funciona para compilar el archivo.java, y después crear el class.

P1>javadoc HolaMundo.java

P1>javac HolaMundo.java

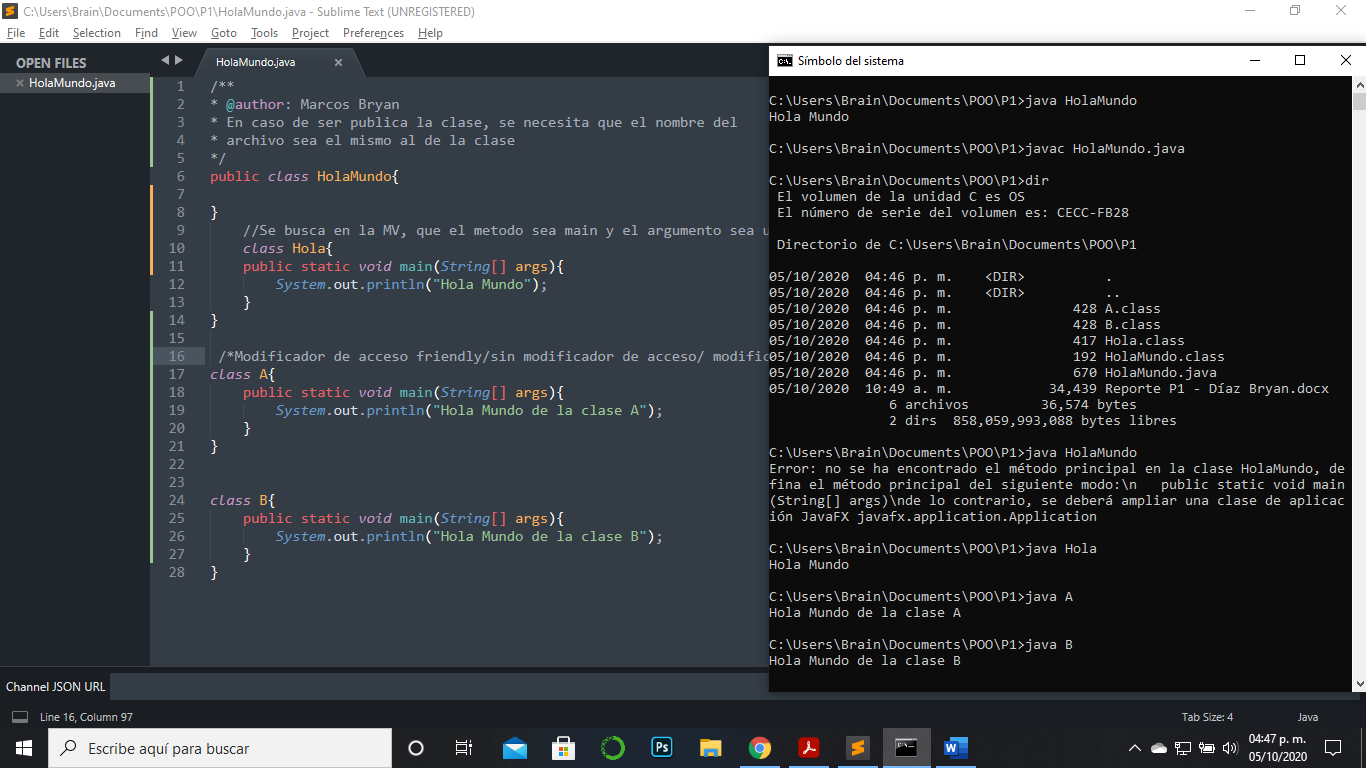
-javadoc: crea la documentación de nuestro programa y guarda la clase y la construcción del programa

* Relación con la teoría / Análisis:

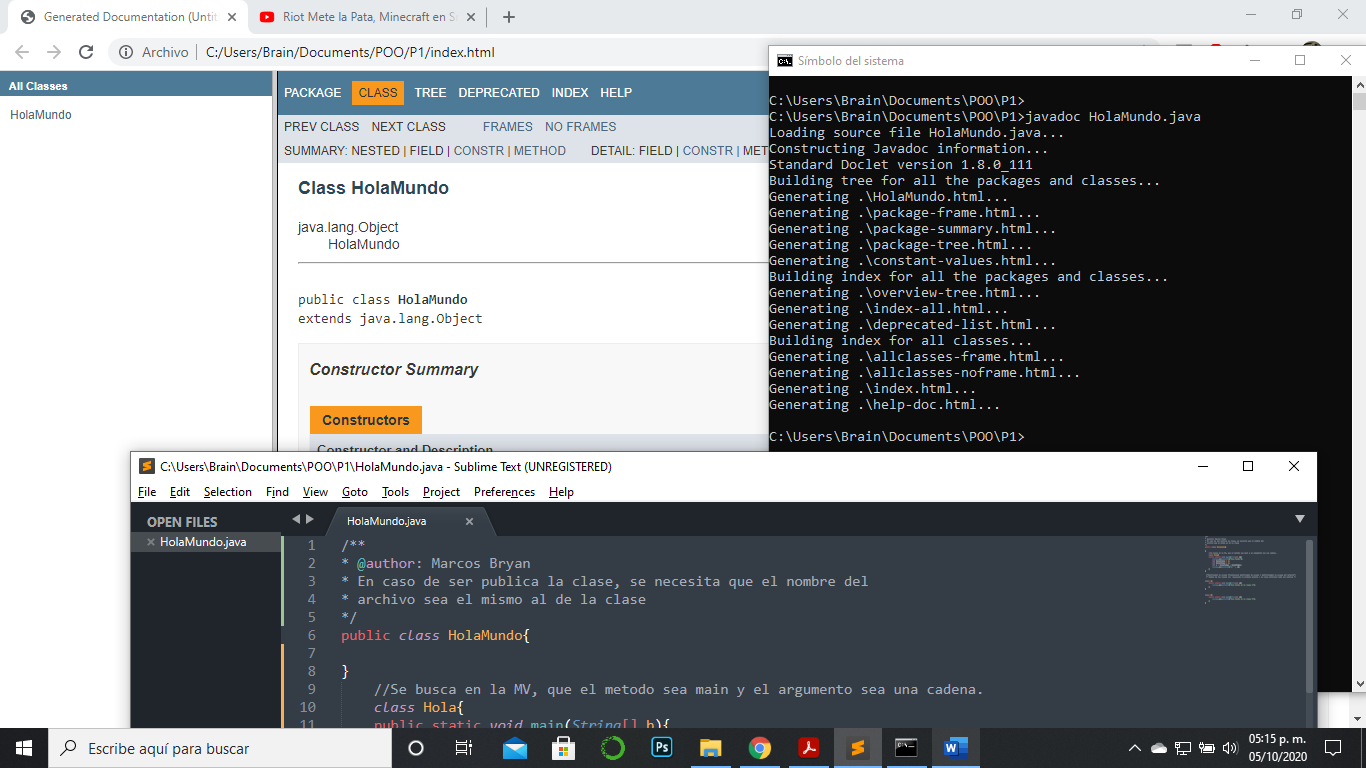
Dentro del programa lo que se realizo fue el implementar los elementos teóricos del paradigma de la Programación Orientada a Objetos, como las clases, los modificadores de acceso, los métodos, y en esencia el comenzar a conocer la interfaz de programación.

En cuanto a la teoría que envuelve el desarrollar esta práctica, termina envolviendo a todo el contenido teórico, sin embargo hay elementos como los objetos, las jerarquías y otros que no utilizamos porque se trato más de empezar a familiarizarnos con el programar, compilar, ejecutar y documentar un programa.

* Evidencias de implementación



En esta parte se muestra la implementación de la compilación del código Java, en la terminal de Windows, además de la ejecución del programa.



En la captura se muestra la implementación de la documentación en la terminal de Java.

**Conclusiones**

En esta práctica he aprendido los elementos básicos de la interfaz de programación de Java, lo que en un momento me resulto algo confuso, al ponerlo en práctica se volvió algo más sencillo de procesar y de aplicar dentro de un lenguaje orientado a objetos. Puedo decir que se cumplió el objetivo de la practica porque pude entender como es el funcionamiento básico de la compilación y ejecución de Java, además de incluso hacer documentación desde la consola, algo que no sabía que se pudiera hacer.

Por último, estoy ansioso de realizar la siguiente practica y poder comprender los otros elementos teóricos que no se vieron en esta práctica, pero que seguro se vayan a estudiar en las próximas, además de poder entender la historia detrás del lenguaje y la forma especial en que este se desarrolla..