



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

<i>Profesor:</i>	Edgar Tista Garcia
<i>Asignatura:</i>	Programación Orientada a Objetos.
<i>Grupo:</i>	Grupo 3
<i>No de Práctica(s):</i>	Práctica 2 – Fundamentos y sintaxis del lenguaje
<i>Integrante(s):</i>	Edwin Jaret Santiago Díaz
<i>No. de Equipo de cómputo empleado:</i>	Trabajo en casa.
<i>No. de Lista o Brigada:</i>	
<i>Semestre:</i>	2022 - 2
<i>Fecha de entrega:</i>	16 febrero 2022
<i>Observaciones:</i>	

CALIFICACIÓN: _____

Fundamentos y sintaxis del lenguaje.

Objetivos

Crear programas que implementen variables y constantes de diferentes tipos de datos, expresiones y estructuras de control de flujo.

Desarrollo

La práctica ya fue finalizada y entregada.

Ejemplos de la guía

Ejemplo 1. Números pares.

En el programa **sumaPares** se le solicita al usuario que escoja 2 números enteros para analizar cuál de los 2 números es el mayor y cuál el menor y así recorrer los números desde el menor hasta el mayor con un ciclo for.

Sumará los números pares que hay entre los números ingresados.

Imprime el número mayor ingresado, el número menor ingresado y la suma de números pares previamente realizada.

```
Por favor introduzca un numero entero
50
Introduzca otro numero entero
28
La suma de los pares entre 50 y 28 es 468
```

Ejemplo 2. Área de una figura.

El programa **FigurasGeometricas** calcula el área de 3 figuras geométricas (círculo, triángulo, cuadrado) dándole la opción a escoger al usuario. Los valores como pi, lado, base, altura el programador los define.

El programa tiene 4 opciones:

1. Calcular el área del círculo (radio = 15)
2. Calcular el área del triángulo (base = 8, altura = 5)
3. Calcular el área del cuadrado (lado = 10)
4. Salir del programa.

El programa me marcó el error “The local variable área may not have been initialized”, investigué y esto se debía que inicialicé la variable **área** y su valor dependería de la opción que el usuario escogiera, sin embargo, existía la posibilidad de que el usuario escogiera cualquier número distinto de 1, 2 o 3 provocando que imprima un valor nulo, por lo tanto, se corrigió asignándole a la variable **área** el valor de 0.

```
1
Selecciono el circulo
El area es: 706.8578
2
Selecciono el triangulo
El area es: 20.0
3
Selecciono cuadrado
El area es: 100.0
4
Su seleccion no es correcta
```

Ejercicios de la clase

Ejercicio 1. Switch-Case

El programa **DiasPorMes** le solicita al usuario que ingrese un mes (enero, marzo o abril) y cuando lo ingresa, el programa debería de imprimir el número de días que tiene ese mes. Estos son los resultados:

```
enero
El mes de enero tiene 28 días
marzo
El mes de marzo tiene 31 días
abril
El mes de abril tiene 31 días
```

En el caso del mes de enero no tiene instrucciones dentro del **case** ni el comando **break**,

esto nos demuestra que el lenguaje **JAVA** compilará y ejecutará el programa sin marcar un error de sintaxis, pero puede llegar a ser el caso de que el resultado deseado no sería el correcto, pues utiliza las instrucciones del siguiente **case**, en este caso las del mes de febrero.

```
enero
El mes de enero tiene 31 dias
noviembre
El mes de noviembre tiene 30 dias
```

Para mejorar el programa, se completan la lista de los meses y en la sentencia de switch asignarle a cada mes cuantos días contienen.

Con esto se demuestra que la sentencia switch-case es ideal para cuando tengamos opciones varias opciones y conforme a cada opción, datos e impresiones distintas.

Ejercicio 2. Operador Ternario

Se creó un programa llamado **OperadorTernario** en donde se tenían que implementar las sentencias if-else en forma de operador ternario. Se cumplió con este objetivo y al ejecutar el programa, esto es lo que nos devuelve:

```
X = 6 Y = 8
El resultado de este operador ternario es: 100
A = 9.0
El resultado del segundo operador ternario es: excelente
```

El primer operador ternario revisa los valores de las variables **X** y **Y**, conforme a sus valores se imprimiría un valor **Z**, los 3 valores son de tipo entero.

En el segundo ternario revisa el valor de la variable **A** que simula una calificación, este valor es de punto flotante, y conforme a su valor se imprimiría un mensaje **B**.

Ejercicio 3. Do-While.

El programa **Adivina el número** es un juego conocido como “Adivina el numero” en el que consiste en el que el programa escoge un número aleatorio del 0 al 10 y el usuario tiene que adivinar ese número y solo tiene 2 oportunidades, el programa le notificará si el numero ingresado es mayor o menor al que debe de adivinar.

El programa fue mejorado para que el número a adivinar sea del rango 0 al 500, el usuario tiene 5 oportunidades y esto se implementó con la sentencia do-while.

Este es un ejemplo cuando el usuario adivina el número:

```
Bienvenido al juego: ADIVINA EL NUMERO
Escribe el numero:
90
El numero que ingresaste es mayor, intenta de nuevo
Escribe el numero:
30
El numero que ingresaste es mayor, intenta de nuevo
Escribe el numero:
13
Felicidades, adivinaste
```

Este es un ejemplo cuando el usuario no adivinó el número

```
Escribe el numero:
432
El numero que ingresaste es mayor, intenta de nuevo
No adivinaste, el numero a adivinar es: 419
```

Ejercicio 4. For-each

La función del programa **ForEach** es recorrer un arreglo de números pares, sumar cada uno de los elementos y guardarlos en una variable, multiplicar cada valor del arreglo por diez y obtener los números que son divisibles entre 4.

El programa fue implementado con ciclos for pero estos fueron cambiados por los ciclos for-each ya que es ideal para trabajar con arreglos. La única limitante de este ciclo (dependiendo de los casos) es que se recorre y obtiene los valores del arreglo y no su índice como lo hace el ciclo for.

El resultado del programa es:



Conclusiones

La sintaxis en el ciclo for, sentencia switch-case, if-else if- else, while y do-while son las mismas que en el lenguaje C, esto genera una gran ventaja para aquellos programadores que saben del lenguaje C. El ciclo for-each es muy útil para recorrer arreglos y tiene una similitud al ciclo for.

Así mismo, no conocía el operador ternario y una vez implementado en el lenguaje Java me resultó mucho más fácil utilizarlo y esto ahorra líneas, el problema de utilizarlo es que al encadenar varios operadores ternarios puede ser algo confuso. Como dato curioso, en esta semana de la entrega de esta práctica realicé un test para Oracle en donde el test fue en java y al conocer el operador ternario me fue más fácil aplicar un algoritmo.

Todos los ejercicios fueron completados y abarcaron los puntos solicitados.