	Carátula para entrega de prácticas
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: TISTA GARCÍA EDGAR

Asignatura: PROGRAMACIÓN ORIENTADA A OBJETOS

Grupo: 3

No de Práctica(s): PRACTICA #2 - FUNDAMENTOS Y SINTAXIS DEL LENGUAJE

Integrante(s): CARRILLO CERVANTES IVETTE ALEJANDRA

No. de Equipo de cómputo empleado: TRABAJO EN CASA

No. de Lista o Brigada: 08

Semestre: 2022 - 1

Fecha de entrega: 22 SEPTIEMBRE 2021

Observaciones:

CALIFICACIÓN: _____

PRACTICA #2 – FUNDAMENTOS Y SINTAXIS DEL LENGUAJE

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

Antes de comenzar la práctica, se activo la página de códigos 65001 desde la terminal, con el fin de que en los programas muestren los caracteres.

EJEMPLOS DE LA GUÍA

Ejemplo 1

Este programa tiene como función principal sumar los números pares comprendidos entre dos valores (los cuales se le piden al usuario), esto se logra a través de la estructura de control "*if-else*" en donde se valida cual es el número mayor y cuál es el número menor ingresado por el usuario, y con base al resultado, se realiza un ciclo "*for*" desde el número menor hasta el número mayor, validando con un "*if*" si es par el número, en dado caso que sea verdadero se va sumando el elemento y guardando en una variable "*suma*" previamente inicializada en cero. Finalmente se imprime el resultado.

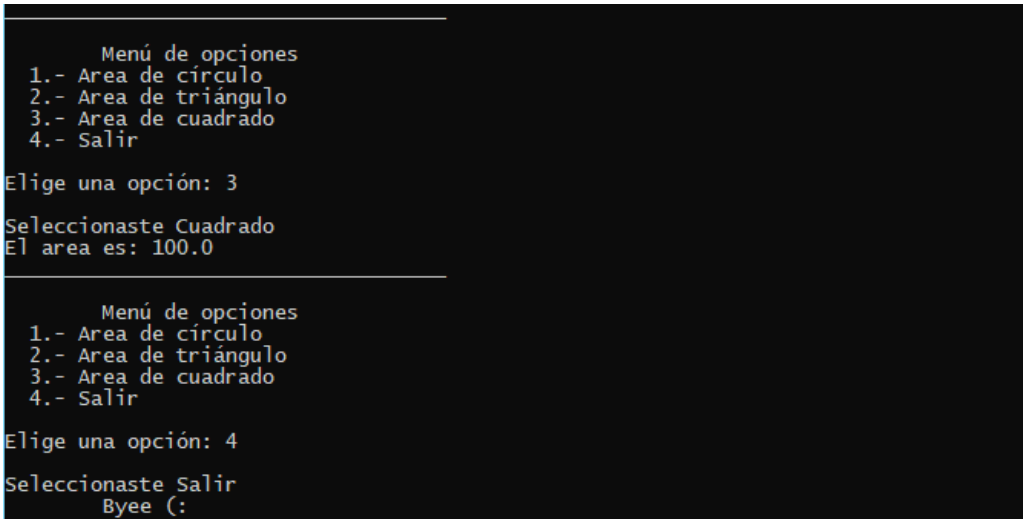
```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1\POO\Prácticas\P2 POO\Carrillo Ce
rvantes Ivette Alejandra G3 P2 V1\Ejemplos>java SumaPares

Por favor introduzca un número entero: 6
Introduzca otro número entero: 16

La suma de los pares entre 6 y 16 es 66 (:
```

./Ejemplos/SumaPares.java

Ejemplo 2

Este programa tiene como función principal calcular el área de una figura geométrica dependiendo la opción seleccionada por el usuario en un menú que se repite hasta que se seleccione la opción "salir". Este programa se realizó con una estructura "*switch-case*" la cual evalúa una variable de tipo int y dependiendo del valor que se selecciona, se hace su instrucción correspondiente:

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1\POO\Prácticas\P2 POO\Carrillo Ce
rvantes Ivette Alejandra G3 P2 V1\Ejemplos>java FigurasGeometricas
--------------------------------

        Menú de opciones
  1.- Area de círculo
  2.- Area de triángulo
  3.- Area de cuadrado
  4.- Salir

Elige una opción: 1

Seleccionaste círculo
El area es: 706.8578
--------------------------------

        Menú de opciones
  1.- Area de círculo
  2.- Area de triángulo
  3.- Area de cuadrado
  4.- Salir

Elige una opción: 2

Seleccionaste Triángulo
El area es: 20.0
```

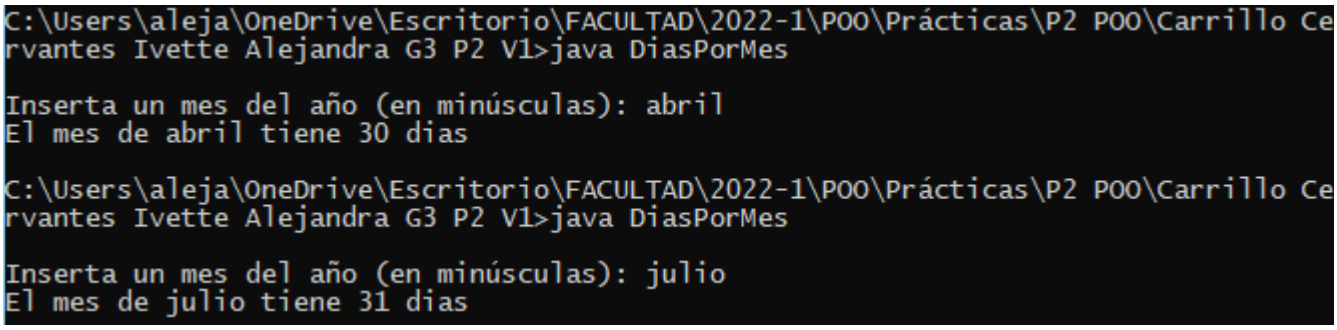


./Ejemplos/FigurasGeometricas.java

ACTIVIDADES PARA EL DESARROLLO DE LA PRÁCTICA

Ejercicio 1. Switch-Case

Este programa tiene como función principal imprimir el número de días de un mes seleccionado por el usuario, para seleccionar dicho mes se utilizó la estructura "*switch-case*" la cual evalúa una variable de tipo String y dependiendo de su valor, se hace realiza su instrucción correspondiente. Finalmente se imprime el mes seleccionado y cuantos días tiene este.



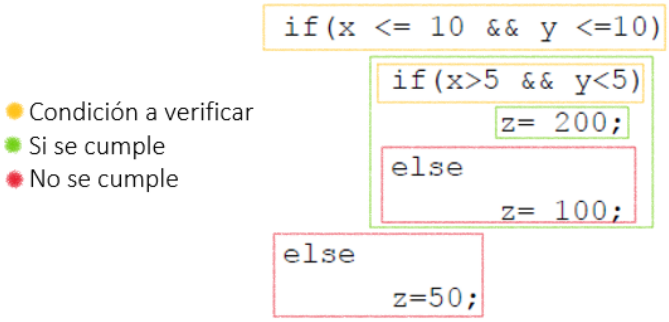
./DiasPorMes.java

Ejercicio 2. Operador ternario

El objetivo que tiene este programa es cambiar la forma de escribir sentencias "*if-else*" por operadores ternarios. En este programa se hizo dicho cambio con dos sentencias "*if-else"*, ambas sentencias tienen "if" anidados y verifican que un número dado por el usuario cumpla ciertas condiciones.

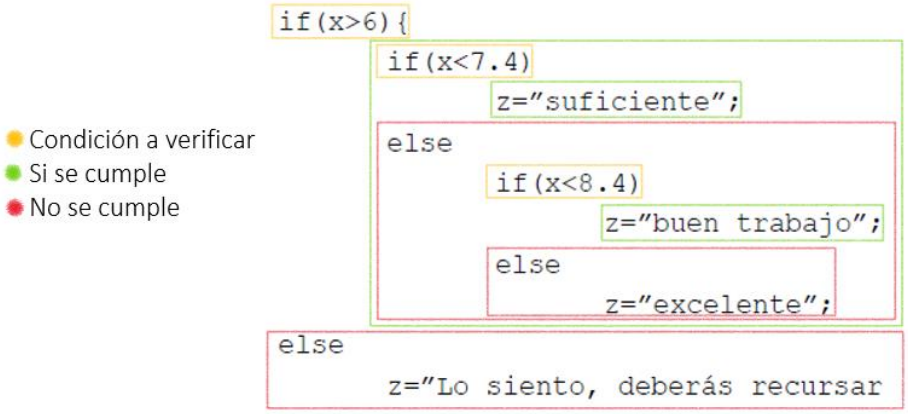
Para ambas sentencias se realizó el siguiente procedimiento con la finalidad de mostrar cada paso que se siguió:

Primera sentencia




```
(x<=10 && y<=10)?(x>5 && y<5)?200:100:50
```

Segunda sentencia




```
(x>6)?(x<7.4)?"suficiente":x<8.4?"buen trabajo":"excelente":"lo siento, deberás recursar"
```

Ya escritas de forma “operador ternario”, se ejecutó dos veces el programa con el fin de comprobar su funcionamiento:

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1\POO\Prácticas\P2 POO\Carrillo Cervantes Ivette Alejandra G3 P2 V1>java OperadorTernario

        OPERADOR TERNARIO

_____________________________________

        PRIMER OPERADOR TERNARIO

Inserta un número x: 7
Inserta un número y: 8
z = 100

_____________________________________

        SEGUNDO OPERADOR TERNARIO

Ingresa tu calificación: 5.6
Lo siento, deberás recursar
```

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1\POO\Prácticas\P2 POO\Carrillo Ce
rvantes Ivette Alejandra G3 P2 V1>java OperadorTernario

        OPERADOR TERNARIO


_____________________________________

        PRIMER OPERADOR TERNARIO

Inserta un número x: 16
Inserta un número y: 6
z = 50


_____________________________________

        SEGUNDO OPERADOR TERNARIO

Ingresa tu calificación: 10
excelente
```

./OperadorTernario.java

Ejercicio 3. Adivina el número

La función principal del programa es adivinar un número creado aleatoriamente en un rango del 0 al 10 (para generar este número aleatorio se debe de importar la clase "*Random"*), para ello solo se tienen 3 intentos, en la implementación de este programa solo se utilizan sentencias "*if-else*" anidadas; sin embargo, se requiere reescribir el código con la estructura "*doWhile"* para que sea más eficiente el programa, asimismo se requiere que ahora el usuario solo tenga 5 intentos para adivinar el número ahora en un rango del 0 al 500, para ello se creó la estructura "*dowhile"* que se va a repetir mientras que una variable previamente inicializada en cero sea menor que 5; dentro de esta estructura se realizó solo una sentencia "*if*-else" en la cual se verifica si el número ingresado es igual, mayor o menor al número por adivinar, en dado caso de que sea mayor o menor se aumenta una variable "*intentos*" previamente inicializada en cero para que al llegar a 5 se salga de la estructura "*doWhile"*; mientras que si el número es igual, otra variable "salir" (también es previamente inicializada en cero y también se va aumentando si el número es mayor o menor), se iguala a 5 e inmediatamente se sale de la estructura "*doWhile"*. Finalmente se imprime el valor del número por adivinar. Para que se vea mejor el funcionamiento de este programa se ejecutó dos veces:

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1\POO\Prácticas\P2 POO\Carrillo Ce
rvantes Ivette Alejandra G3 P2 V1>java AdivinaNumero

        Bienvenido al juego 'ADIVINA EL NÚMERO'

// El número es: 217

  -> Intento 1
Escribe el numero: 200
El número que ingresaste es MENOR ):

  -> Intento 2
Escribe el numero: 400
El número que ingresaste es MAYOR ):

  -> Intento 3
Escribe el numero: 300
El número que ingresaste es MAYOR ):

  -> Intento 4
Escribe el numero: 250
El número que ingresaste es MAYOR ):

  -> Intento 5
Escribe el numero: 210
El número que ingresaste es MENOR ):

NO ADIVINASTE!!
El número era: 217
```

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1\POO\Prácticas\P2 POO\Carrillo Cervantes Ivette Alejandra G3 P2 V1>java AdivinaNumero

       Bienvenido al juego 'ADIVINA EL NÚMERO'

// El número es: 36

  -> Intento 1
Escribe el numero: 50
El número que ingresaste es MAYOR ):

  -> Intento 2
Escribe el numero: 36

FELICIDADES, ADIVINASTE!! (:
El número es: 36
```

./AdivinaNumero.java

Ejercicio 4. Foreach

Este programa tiene como objetivo reescribir 3 estructuras de repetición "*For"* a 3 estructuras de repetición "*ForEach*", dichas estructuras tienen las siguientes funciones:

- Suma de elementos

En cada iteración se va sumando el elemento con el elemento anterior.

- Cambiar valores de los elementos

Se va multiplicando cada elemento por 10 y se guarda en su indice correspondiente.

- Imprimir los números que son divisibles entre 4

Recorre el arreglo y verifica cada elemento, en caso de que dicho elemento sea divisible entre 4 se imprime.

Para verificar que el programa cumple con su función, se imprimió primero las estructuras de repetición "*For"* y después las estructuras de repetición "*ForEach*", se observa que ambas estructuras dan el mismo resultado.

```
_ _ _ _ _ _ _ _ _ _ _ _ FOR _ _ _ _ _ _ _ _ _ _ _ _

_ _ _ _ _ _ _ _ _ _ _ _

  SUMA DE ELEMENTOS

5
15
30
50
75
105
140
180
225
275

_ _ _ _ _ _ _ _ _ _ _ _

  CAMBIAR VALORES DE ELEMENTOS

50
100
150
200
250
300
350
400
450
500

_ _ _ _ _ _ _ _ _ _ _ _

  NUMEROS DIVISIBLES ENTRE 4

100 es divisible entre 4
200 es divisible entre 4
300 es divisible entre 4
400 es divisible entre 4
500 es divisible entre 4

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1
\POO\Prácticas\P2 POO\Carrillo Cervantes Ivette Al
ejandra G3 P2 V1>
```

```
_ _ _ _ _ _ _ _ _ _ _ _ FOR EACH _ _ _ _ _ _ _ _ _ _ _ _

_ _ _ _ _ _ _ _ _ _ _ _

  SUMA DE ELEMENTOS

5
15
30
50
75
105
140
180
225
275

_ _ _ _ _ _ _ _ _ _ _ _

  CAMBIAR VALORES DE ELEMENTOS

50
100
150
200
250
300
350
400
450
500

_ _ _ _ _ _ _ _ _ _ _ _

  NUMEROS DIVISIBLES ENTRE 4

100 es divisible entre 4
200 es divisible entre 4
300 es divisible entre 4
400 es divisible entre 4
500 es divisible entre 4

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1
\POO\Prácticas\P2 POO\Carrillo Cervantes Ivette Al
ejandra G3 P2 V1>
```

./ForEach.java

Conclusiones

Se cumplieron con los objetivos de esta práctica ya que se implementaron variables de diferentes tipos de datos, como: String, int, float; al igual que se implementaron expresiones y estructuras de control de flujo como doWhile, switch, for, foreach, al igual que el operador ternario.

Se realizaron todos los ejercicios de la práctica; sin embargo, me causo un poco de conflicto el ejercicio donde se implementaba el operador ternario, ya que fue un poco confuso donde terminaba una instrucción y empezaba otra, pero al dibujarlo ya me quedo más claro ese concepto. Con respecto a las estructuras de control, al principio pensé que su sintaxis era diferente a la de C; sin embargo, funciona exactamente igual.

Considero que los ejercicios propuestos para la práctica fueron adecuados para los temas vistos en clase, ya que apenas vamos empezando a aprender el lenguaje JAVA. (: