Instituto Tecnológico y de Estudios

Superiores de Occidente – ITESO



Materia: Programación Orientada a Objetos

Maestro: Hugo Iván Piza Dávila

TAREA 1

Sesión: 4

Fecha: 29 de agosto el 2025

Temas: Condicionales  
 Control de Flujo

Autor: Alor Santiago Oscar Alberto

PROBLEMA 1

Descripción:

Determina el signo zodiacal de una persona a partir de su fecha de nacimiento.

Código fuente:

Despliega dos cuadros de diálogo de entrada de datos. La primera entrada solicita “Mes de nacimiento” mientras que el segundo solicita el “día de nacimiento”:

|  |  |
| --- | --- |
| 1  2 | String mes = JOptionPane.showInputDialog("Dame tu mes de nacimiento");  String diaNacimiento = JOptionPane.showInputDialog("Dame tu dia de nacimiento"); |

Se crea un nuevo String “mesZodiacal” para el mes el (Se pudo sobrescribir el primero realmente) en donde se modifica el contenido, primero crean una cadena nueva que elimina espacio y después un nuevo objeto que ahora es todo minúscula. después se crear un int “diaNac” el cual guardará la conversión de String a Int de “díaNacimiento”.

|  |  |
| --- | --- |
| 1  2 | String mesZodiacal = mes.trim().toLowerCase();  **int** diaNac = Integer.parseInt(diaNacimiento); |

después se ejecuta el código principal el cual es la asignación de un string “signoZodiacal” que guardará el sigo dependiendo de la condición. Para las condiciones solo se tomo en cuenta que la primera mitad del mes corresponde a un signo Zodiacal mientras que para la otra mitad es diferente, el siguiente mes inicia su primera mitad con el ultimo Signo Zodiacal del mes anterior y la otra mitad cambia y así sucesivamente.

Nótese que el switch evalúa el contenido de mesZodiacal que fue modificado anteriormente:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | String signoZodiacal = **switch** (mesZodiacal){  **case** "enero" -> (diaNac < 20) ? "Capricornio" : "Acuario";  **case** "febrero" -> (diaNac < 19) ? "Acuario" : "Piscis";  **case** "marzo" -> (diaNac < 20) ? "Piscis" : "Aries";  **case** "abril" -> (diaNac < 20) ? "Aries" : "Tauro";  **case** "mayo" -> (diaNac < 21) ? "Tauro" : "Geminis";  **case** "junio" -> (diaNac < 21) ? "Geminis" : "Cancer";  **case** "julio" -> (diaNac < 23) ? "Cancer" : "Leo";  **case** "agosto" -> (diaNac < 23) ? "Leo" : "Virgo";  **case** "septiembre" -> (diaNac < 23) ? "Virgo" : "Libra";  **case** "octubre" -> (diaNac < 23) ? "Libra" : "Escorpio";  **case** "noviembre" -> (diaNac < 22) ? "Escorpio" : "Sagitario";  **case** "diciembre" -> (diaNac < 22) ? "Sagitario" : "Capricornio";  **default** -> "Mes no valido";  }; |

así mismo el valor por default es “Mes no valido”, en la última parte del código se evalúa si el contenido de “signoZodiacal” es igual a esta cadena de caracteres, en caso de que si, se envía dicho mensaje al usuario, de lo contrario, se lanza un nuevo mensaje con todos los valores del usuario, así como el resultado:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | *// Conditional*  **if** (signoZodiacal.equals("Mes no valido")) {  JOptionPane.showMessageDialog(**null**, signoZodiacal);  }**else** {  JOptionPane.showMessageDialog(**null**, String.format("Dado que naciste el %d de %s tu signo zodiacal es %s", diaNac, mesZodiacal, signoZodiacal));  } |

Ejecución:

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto. Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto. Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto. Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto. Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

PROBLEMA 2

Descripción:

Encuentra el número primo más grande en un rango [A, B] dado por el usuario.

Código fuente:

Se despliegan dos cuadros de diálogos, el primero solicita un numero que será “menor” en el rango y el segundo será el numero “mayor” del mismo rango.

|  |  |
| --- | --- |
| 1  2 | String minNumero = JOptionPane.showInputDialog("Escribe el numero menor");  String maxNumero = JOptionPane.showInputDialog("Escribe el numero mayor"); |

Despues de ello se “parsea” cada “String” para se guardado dentro de un “int” que será usado para la evaluación.

|  |  |
| --- | --- |
| 1  2 | **int** minNum = Integer.parseInt(minNumero);  **int** maxNum = Integer.parseInt(maxNumero); |

Después de ello se declaran dos variables. Una es “control” que usada dentro del ciclo “while” para confirmar si dentro de las divisiones posibles de un número, hubo alguna operación que dividió el numero siendo evaluado y quedó en cero, si ello sucede “control” se queda con el valor de “0” y se rompe el ciclo. El otro valor es “result” el cual se encarga de almacenar el valor del último primo encontrado dependiendo de lo que dé el resultado del “while” al finalizar (véase más adelante).

|  |  |
| --- | --- |
| 1  2 | **int** control = 0;  **int** result = 0; |

La lógica del “for” solamente itera entre cada número del rango y en cada iteración reinicia el valor de “divisor”. El “while” por otro lado siempre se ejecuta al menos una vez y tiene dos condiciones, si el numero evaluado al ser dividido entre el “divisor” es igual a 0 alguna vez, significa que no es primo, nótese que se empieza en 2 y termina en cuando divisor es igual a la raíz cuadrada del numero evaluado en el “for”.

Asi mismo, si el resultado es diferente de 0 entonces el divisor aumenta (Estando condicionado por el “while”) y seguirá aumentando siempre y cuando no haya resultados que den 0 (Si hay no es primo) o cuando se alcance la raíz cuadrada (Es primo).

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | **for**(**int** i = minNum; i <= maxNum; i++) {  **int** divisor = 2;  **do** {  **if**( i%(**double**)divisor == 0 ) {  control = 0;  **break**;  }  divisor++;  control = 1;  }**while**(divisor <= (**double**)Math.sqrt(i));  *// Acaba While*  **if** (control == 1) result = i;  } |

Aquí sucede lo siguiente:

1. Si la división del numero siendo evaluado por la iteración NUNCA da cero, entonces es primo y el valor se guarda en “result”, el cual ahora tiene el numero primo. Cambiará solo si encuentra otro dentro del rango (El mayor).
2. Si la división alguna vez da cero, “control” es 0, se termina el programa y aquí suceden dos cosas también:
   1. Dado que “control” finaliza en cero, “result” queda en 0 como se declaró originalmente lo que significa que no es primo porque alguna vez se dividió correctamente.
   2. Si la división nunca quedó en 0, entonces “control” es 1 siempre y la condicional fuera del “while” asigna el valor de la iteración a “result”.

La explicación de arriba queda mas clara con la última parte del código en el cual avisa que no hay número primos si “result” se mantuvo en 0 o bien, avisa del último primo guardado si alguna vez algún número del rango cumplió con que ninguna división dio cero:

|  |  |
| --- | --- |
| 1  2  3  4  5 | **if**(result == 0) {  JOptionPane.showMessageDialog(**null**, "No hay numero primos en tu rango");  }**else** {  JOptionPane.showMessageDialog(**null**, String.format("El primo numero mas grande entre %d y % d es %d", minNum, maxNum, result));  } |

Ejecución:

Para la ejecución se usaron números que abarquen el siguiente rango de primos, los primos dos deben dar como resultado 43 y 97 mientras que los otros dos están contenidos entre [“479+1” - “487-1”] y [“953+1” - “967-1”] no deben arrojar resultados:

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

NO PRIMOS ENCONTRADOS AHORA:

Interfaz de usuario gráfica

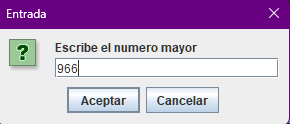
El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

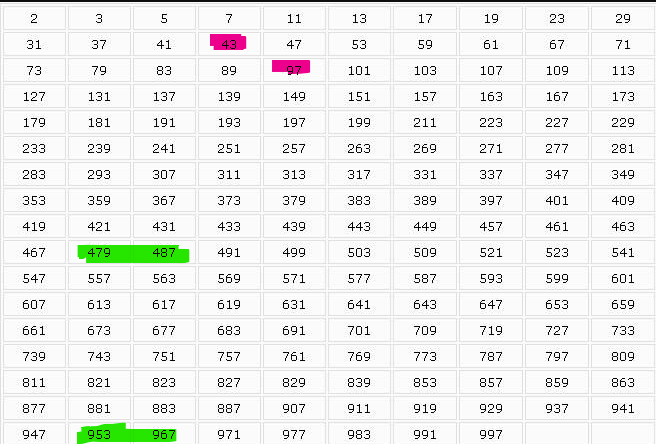
El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.



Codigo Fuente: <https://github.com/OsrKozuki/POO_ESI017O/blob/main/startingPoo/src/tareas/tareaUno.java>