|  |
| --- |
| **FUNDAMENTOS DE LA PROGRAMACION ORIENTADO A OBJETOS** |

**Trabajo práctico/Actividad**

**N°1**

**Apellido y Nombre – LU/**

Calatayud, Alex Gabriel

TUV000576

**Grupo:**

**Integrantes**

**AyN /LU**

**Profesor:**

Mg. Ing. Ariel Alejandro Vega

**Año: 2024**

Indice

**Sección Expresiones aritméticas y lógicas**

Resolver cada ejercicio en un archivo Word y luego programarlo en Processing. En el caso de la programación crear un archivo por ejercicio.

**Ejercicio 1:** Evaluar (obtener resultado) la siguiente expresión para A = 2 y B = 5

**3\* A - 4 \* B / A ^ 2**

Resolución necesaria en Word:

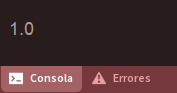
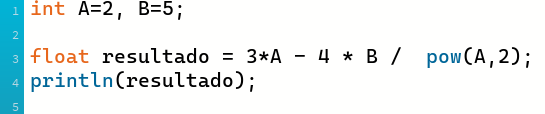
(3\*A)-(4\*B/(A^2))

6-(4\*B/4)

6-5

1

**Captura de Processing** **/ Resultado:**



**Ejercicio 2:** Evaluar la siguiente expresión:

**4 / 2 \* 3 / 6 + 6 / 2 / 1 / 5 ^ 2 / 4 \* 2**

|  |  |
| --- | --- |
| Aritmética | Algebraica |
| **4 / 2 \* 3 / 6 + 6 / 2 / 1 / 5 ^ 2 / 4 \* 2**  (((4/2) \* 3) /6 ) + ((((6/2) / 1) / (5 ^ 2)) / 4) \* 2  1.0 + 0.06  1.06 |  |

|  |  |
| --- | --- |
| Captura processing | Resultado |
|  |  |

**Ejercicio 3**: Evaluar las siguientes expresiones aritméticas, para lo cual indicar en el caso de las variables, el valor indicado. Luego escribirlas como expresiones algebraicas.

**a)**

**b ^ 2 – 4 \* a \* c**  a= 2, b=4, c=1

|  |  |
| --- | --- |
| Aritmética | Algebraica |
| **b ^ 2 – 4 \* a \* c**  (4 ^ 2) - (4 \* 2 \* 1)  16 – 8  8 |  |

|  |  |
| --- | --- |
| Captura de processing | Resultado |
|  |  |

**b)**

**3 \* X ^ 4 – 5 \* X ^ 3 + X 12 – 17** x=5

|  |  |
| --- | --- |
| Aritmética | Algebraica |
| **3 \* X ^ 4 – 5 \* X ^ 3 + X 12 – 17**  (3\*(5^4)) - (5 \* (5^3)) + (5\*12) - 17  1875 – 625 + 60 – 17  1293 |  |

|  |  |
| --- | --- |
| Captura processing | Resultado |
|  |  |

**c)**

**(b + d) / (c + 4)** b=2, c=1, d=4

|  |  |
| --- | --- |
| Aritmética | Algebraica |
| **(b + d) / (c + 4)**  (2 + 4) / (1 + 4)  1.2 |  |

|  |  |
| --- | --- |
| Captura processing | Resultado |
|  |  |

**d)**

**(x ^ 2 + y ^ 2) ^ (1 / 2)** x=2 y=3

|  |  |
| --- | --- |
| Aritmética | Algebraica |
| **(x ^ 2 + y ^ 2) ^ (1 / 2)**  (2^2 + 3^2) ^ (1 / 2)  13 ^ (1 / 2)  3.605512 |  |

|  |  |
| --- | --- |
| Captura processing | Resultado |
|  |  |

**Ejercicio 4:** Si el valor de A es 4, el valor de B es 5 y el valor de C es 1, evaluar las siguientes expresiones:

a)

**B \* A – B ^ 2 / 4 \* C**

|  |  |
| --- | --- |
| Aritmética | Algebraica |
| **B \* A – B ^ 2 / 4 \* C**  5 \* 4 – ((5 ^ 2) / 4) \* 1  20 - 6.25  13.75 |  |

|  |  |
| --- | --- |
| Captura processing | Resultado |
|  |  |

**b)**

**(A \* B) / 3 ^ 2**

|  |  |
| --- | --- |
| Aritmética | Algebraica |
| **(A \* B) / 3 ^ 2**  (4 \* 5) / 3 ^ 2  2.2 ... |  |

|  |  |
| --- | --- |
| Captura processing | Resultado |
|  |  |

**c)**

**(((B + C) / 2 \* A + 10) \* 3 \* B) – 6**

|  |  |
| --- | --- |
| Aritmética | Algebraica |
| **(((B + C) / 2 \* A + 10) \* 3 \* B) – 6**  (((5 + 1) / 2 \* 4 + 10) \* 3 \* 5) - 6  ((6 / 2 \* 4 + 10) \* 3 \* 5) - 6  ((3 \* 4 + 10) \* 3 \* 5) - 6  (22 \* 3 \* 5) - 6  330 – 6  324 |  |

|  |  |
| --- | --- |
| Captura processing | Resultado |
|  |  |

**Ejercicio 5:** Para x=3, y=4; z=1, evaluar el resultado de:

R1 = y+z

R2 = x >= R1

R1= 4+1 = 5

R2= 3 >= R1

Falso

|  |  |
| --- | --- |
| Captura processing | Resultado |
|  |  |

**Ejercicio 6:** Para contador1=3, contador2=4, evaluar el resultado de

R1 = ++contador1

R2 = contador1 < contador2

R2= 4 < 4

R2= falso

|  |  |
| --- | --- |
| Captura processing | Resultado |
|  |  |

**Ejercicio 7:** Para a=31, b=-1; x=3, y=2, evaluar el resultado de a+b-1 < x\*y

a+b-1 < x\*y

31+(-1)-1 < 3\*2

29 < 12

falso

|  |  |
| --- | --- |
| Captura processing | Resultado |
|  |  |

**Ejercicio 8:** Para x=6, y=8, evaluar el resultado de ! (x<5) &&!(y>=7)

!(x<5) && !(y>=7)

!(6<5) && !(8>=7)

falso && falso

falso

|  |  |
| --- | --- |
| Captura processing | Resultado |
|  |  |

**Ejercicio 9:** Para i=22, j=3, evaluar el resultado de !((i>4) || !(j<=6))

!((i>4) || !(j<=6))

!((22>4) || !(3<=6))

!(verdadero || falso)

!(verdadero)

falso

|  |  |
| --- | --- |
| Captura processing | Resultado |
|  |  |

**Ejercicio 10:** Para a=34, b=12, c=8, evaluar el resultado de !(a+b==c) || (c!=0) && (b-c>=19)

!(a+b==c) || (c!=0) && (b-c>=19)

!(34+12==8) || (8!=0)&&(12-8>=19)

!(46==8) || (8!=0)&&(4>=19)

verdadero || verdadero && falso

verdadero|| falso

verdadero

|  |  |
| --- | --- |
| Captura processing | Resultado |
|  |  |

**Sección Análisis – Diseño y Codificación de algoritmos – Aplicación de estructuras de control**

Para cada ejercicio, en el archivo Word agregar las secciones de análisis y diseño, mientras que, para la codificación, crear el archivo de Processing

**Ejercicio 11:** Un problema sencillo. Deberá pedir por teclado al usuario un nombre y posteriormente realizará la presentación en pantalla de un saludo con el nombre indicado.

**Análisis:**

Datos de Entrada: nombre\_ingresado //cadena

Datos de Salida: mensaje\_saludo //cadena de texto

Proceso:

**¿Quien debe realizar el proceso?:** El algoritmo o computadora

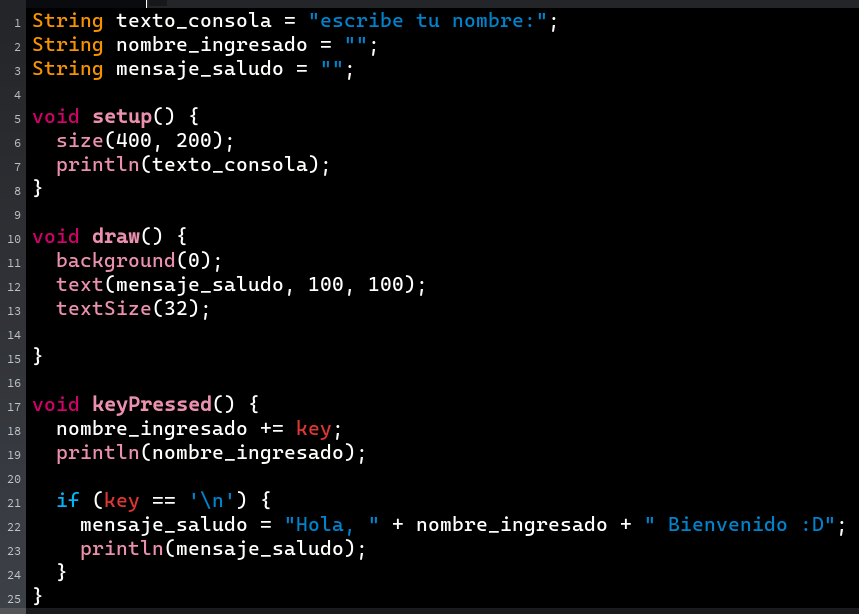
**¿Cual es el proceso que resuelve?:** Ingresar un nombre que devolverá la creación de un saludo personalizado con el nombre proporcionado y su presentación en pantalla.

**Diseño:**

|  |
| --- |
| **Entidad que resuelve el problema: Algoritmo** |
| **Variables:**  **nombre\_ingresado: string //** almacena el nombre  **mensaje\_saludo: string //** almacenara una cadena de caracteres |

|  |
| --- |
| **Nombre del Algoritmo:** saludar\_nombre |
| **Proceso del algoritmo:**   1. *inicio* 2. *Leer nombre\_ingresado* 3. *mensaje\_saludo ← “Hola, ” + nombre\_ingresado + “ ¡Bienvenido!”* 4. *Mostrar saludo* 5. *fin* |

Captura de la codificación en lenguaje processing



Resultado en la terminal:

|  |  |  |
| --- | --- | --- |
| Mensaje que aparece al iniciar el programa. | Al escribir, el programa va escuchando cada tecla presionada y finalmente al presionar la tecla **enter,** lanza el mensaje de saludo. | En la ventana del dibujo se muestra lo siguiente: |

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Ejercicio 12:** Será común resolver problemas utilizando variables. Calcule el perímetro y área de un rectángulo dada su base y su altura.

**Análisis:**

Datos de Entrada: base, altura //decimal

Datos de Salida: perimetro, area // almacena valores decimales

Proceso:

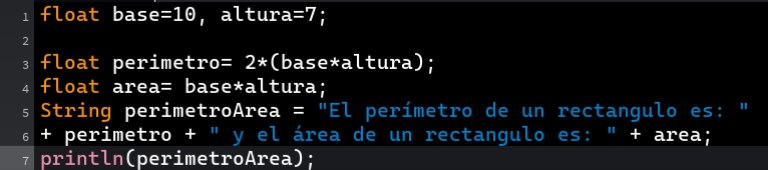
**¿Quién debe realizar el proceso?:** El usuario o calculadora

**¿Cuál es el proceso que resuelve?:** calcula el perímetro y el área de un rectángulo utilizando las fórmulas adecuadas

**Diseño:**

|  |
| --- |
| **Entidad que resuelve el problema: persona** |
| **Variables:**   * **base: float //** almacena un valor decimal * **area: float** // almacena un valor decimal * **perimetro: float //** * **area: float //** * **perimetroArea: float // almacena un valor de calculos** |

|  |
| --- |
| **Nombre del Algoritmo:** **perimetro\_area\_rectangulo** |
| **Proceso del algoritmo:**   1. *inicio* 2. *Leer base* 3. *Leer area* 4. *perimetro ← 2\*(base + altura)* 5. *area ← base \* altura* 6. *perimetroArea ← “el perimetro de un rectángulo es: ” + perimetro + “ y la area de un rectángulo es: ” + area* 7. *Mostrar perimetroArea* 8. *fin* |

Captura de la codificación en lenguaje processing

|  |
| --- |
| Resultado en la consola |
|  |

**Ejercicio 13**: Una ayuda importante al momento de resolver problemas con algoritmos es asumir que su gran amigo son las matemáticas. Obtenga la hipotenusa de un triángulo rectángulo conociendo sus catetos

**Análisis:**

Datos de Entrada: catetoA, catetoB

Datos de Salida: hipotenusa

Proceso:

**¿Quien debe realizar el proceso?:** La persona o calculadora

**¿Cual es el proceso que resuelve?:** Para calcular la longitud de la hipotenusa de un triángulo rectángulo se obtiene las longitudes de los catetos como entrada, se aplica la fórmula:

y se muestra el resultado como salida.

**Diseño:**

|  |
| --- |
| **Entidad que resuelve el problema: persona** |
| **Variables:**   * **catetoA: int //** almacena un valor decimal * **catetoB: int // almacena un valor decimal** * **hipotenusa: int //** almacena un valor de calculos |

|  |
| --- |
| **Nombre del Algoritmo:** **perimetro\_area\_rectangulo** |
| **Proceso del algoritmo:**  *Leer catetoA*   1. *inicio* 2. *Leer catetoB* 3. *hipotenusa ← (a^2 + b^2 ) ^(0.5)* 4. *mostrar hipotenusa* 5. *Fin* |

|  |  |
| --- | --- |
| Captura de la codificación en lenguaje processing | Resultado en la terminal |
|  |  |

Conclusión

Párrafos de las conclusiones

Fuentes bibliográficas

Se deben enunciar las fuentes (apuntes de la materia, páginas web, videos de youtube, libro (nombre, autores, año), etc)