|  |  |  |
| --- | --- | --- |
| Texto  Descripción generada automáticamente con confianza media | TECNICATURA UNIVERSITARIA EN DISEÑO INTEGRAL DE VIDEOJUEGOS  FACULTAD DE INGENIERÍA  Universidad Nacional de Jujuy |  |

**FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS**

Trabajo Práctico/Actividad

N.º 1

Leaño Dario Gonzalo Alejandro –

LU: TUV000227

*Profesores:*

*Mg. Ing. Ariel Alejandro Vega*

*Ing. Carolina Cecilia Apaza*

*Año*

Indice

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

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

*3\*2 – 4\*5 / 2²*

*6 – 20/4*

*6 – 5*

***1***

Ejercicio 2: Evaluar la siguiente expresión 4 / 2 \* 3 / 6 + 6 / 2 / 1 / 5 ^ 2 / 4 \* 2

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

(2\*3/6)+(3/1/25/4\*2)

(6/6)+(3/25/4\*2)

1 + (0.12/4\*2)

1 + (0.03\*2)

1 + 0.06

**1.06**

Ejercicio 3: Escribir las siguientes expresiones algebraicas como expresiones algorítmicas (en su forma aritmética dentro del algoritmo). En este caso no se pide evaluarlas ni programarlas.

IGNORAR EJERCICIO

Ejercicio 4: Evaluar las siguientes expresiones aritméticas, para lo cual indicar en el caso de las variables, el valor indicado. **Luego escribirlas como expresiones algebraicas.**

1. b ^ 2 – 4 \* a \* c
2. 3 \* X ^ 4 – 5 \* X ^ 3 + X 12 – 17
3. (b + d) / (c + 4)
4. (x ^ 2 + y ^ 2) ^ (1 / 2)

Para aclarar que indicamos con ”Luego escribirlas como expresiones algebraicas” lo aplicamos con el punto a)

*𝑏2 − 4. 𝑎. 𝑐*

*Si A=2, B=5, C= 3, D=9, X=1; Y=4*

*a)*

*B² – 4\*A\*C*

*(5² )– (4x2x3)*

*25 – 24*

*1*

*b)*

1. 3 \* X ⁴ – 5 \* X³ + X\*12 – 17
2. 3\*1² – 5\*1³ + 1\*12 – 17
3. 3 – 5 + 12 – 17
4. -2 +12 -17
5. 10 -17
6. -7

*c)*

(b + d) / (c + 4)

(5+9)/(3+4)

14/7

*d)*

1. (x ^ 2 + y ^ 2) ^ (1 / 2)
2. (1² + 4² )^(1/2)
3. (1+16)^(1/2)
4. 17^(1/2)
5. 4.1231055

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

1. a) B \* A – B ^ 2 / 4 \* C
2. (5\*4) – ((5²) /4\*1)
3. 20 – (25/4)
4. 13,75
5. b) (A \* B) / 3 ^ 2
6. (4\*5)/(3²)
7. 20/9
8. 2.2222223
9. c) (((B + C) / 2 \* A + 10) \* 3 \* B) – 6
10. (((5+1)/2\*4+10)\*3\*5)-6
11. ((6/2\*4+10)\*3\*5)-6
12. ((3\*4+10)\*3\*5)-6
13. ((12+10)\*3\*5)-6
14. (22\*3\*5)-6
15. (66\*5)-6
16. 330-6
17. 324

Ejercicio 6: Para x=3, y=4; z=1, evaluar el resultado de R1 = y+z

R2 = x >= R1

R1=4+1

R1=5

R2=3>=R1

La expresion es falsa

Ejercicio 7: Para contador1=3, contador2=4, evaluar el resultado de R1 = ++contador1

R2 = contador1 < contador2

R1>R2

La expresion es falsa

r1=r2

Ejercicio 8: 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 < 6

La expresion es falsa: 29>6

Ejercicio 9: Para x=6, y=8, evaluar el resultado de

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

!(6<5) (Y) !(8≥7)

!(falso) (Y) !(verdadero)

verdadero (Y) falso

falso

Ejercicio 10: Para i=22, j=3, evaluar el resultado de

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

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

!((verdadera) || !(verdadera))

!(verdadera || falsa)

!(verdadera)

falsa

Ejercicio 11: Para a=34, b=12,c=8, evaluar el resultado de

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

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

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

!(falso) || (verdadero) && (falso)

verdadero || verdadero && falso

verdadero || falso

verdadero

**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 12: 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.

**Definición del Problema**: Ingresar un nombre y saludarlo

**Análisis**:

* Datos de Entrada: nombre de la persona
* Datos de Salida: “Buen día ……” …. :nombre de la persona
* Proceso:

¿Quién debe realizar el proceso?: Un usuario

¿Cuál es el proceso que realiza …? Unir el mensaje en conjunto con el nombre del usuario

**Diseño**:

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA**: Usuario |
| **VARIABLES**  nombreUsuario: String // almacena el nombre del usuario que se ingresa  mensajeBienvenida: String // almacena el mensaje de saludo final |
| **NOMBRE ALGORITMO**: saludar\_al\_usuario  **PROCESO DEL ALGORITMO**   1. *Mostrar* “Ingrese su nombre: ” 2. *Leer* nombreUsuario 3. mensajeBienvenida ← “Buen día ” + nombreUsuario 4. *Mostrar* mensajeBienvenida |

Ejercicio 13: Será común resolver problemas utilizando variables. Calcule el perímetro y área de un rectángulo dada su base y su altura.

**Definición del problema:** Calcular el perímetro y el área del rectángulo

**Análisis**:

* Datos de Entrada: base, altura
* Datos de Salida: perímetro, área
* Proceso:

¿Quién debe realizar el proceso?: El rectángulo

¿Cuál es el proceso que realiza …?

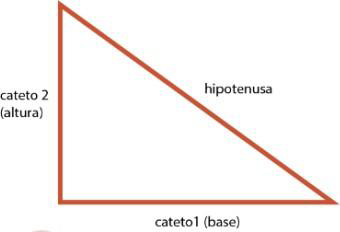
1° Perímetro: sumar la base y el área y multiplicarlo por 2: (base\*altura)\*2

2° Área: multiplicar la base y la altura: base\*altura

**Diseño**:

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA**: Rectángulo |
| **VARIABLES**  *baseRect*: Real // almacena el número que se ingresa  *alturaRect*: Real// almacena el número que se ingresa  *perimetroRect*: Real// almacena el perímetro del rectangulo  *AreaRec*t: Real// almacena el área del rectángulo |
| **NOMBRE ALGORITMO**: calcular\_perimetro\_rectangulo  **PROCESO DEL ALGORITMO**   1. *Leer* *baseRect* 2. *Leer* *alturaRect* 3. *perimetroRect* ← (baseRect + *alturaRect)* \* 2 4. *areaRect* ← *baseRect* \* *alturaRect* 5. *Mostrar perimetroRect* 6. Mostrar *areaRect* |

Ejercicio 14: 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



**Definición del Problema**: Obtener la hipotenusa de un triángulo rectángulo

**Análisis**:

* Datos de Entrada:

Base: Real

Altura: Real

* Datos de Salida:

Hipotenusa: Real

* Proceso:

¿Quién debe realizar el proceso?:

Una calculadora

¿Cuál es el proceso que realiza la calculadora?

Se realiza el calculo:

Hipotenusa^2=(base^2+altura^2)

**Diseño**:

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA**: Calculadora |
| **VARIABLES**  base, altura: Real // almacenan las variables  hipotenusa: Real // almacena el resultado |
| **NOMBRE ALGORITMO**: sacar\_hipotenusa\_triangulo\_rectangulo  **PROCESO DEL ALGORITMO**   * + - 1. *Leer* base       2. *Leer* altura       3. hipotenusa ← (base^2 + altura^2) \* (1/2)       4. *Mostrar* hipotenusa |

Ejercicio 15: Si viste algo de los apuntes y vídeos, esto debería ser muy fácil de resolver. Dados dos números permita calcular la suma, resta, multiplicación y división de estos. Considere que cada una de estas operaciones es un algoritmo cuando realice el diseño. Obviamente muestre los resultados.

**Definición del Problema**: determinar un algoritmo que permita sumar, restar, multiplicar y dividir dos números.

**Análisis**:

* Datos de Entrada:

numeroA, numeroB: Real

* Datos de Salida:

suma

resta

multiplicación

división

* Proceso:

¿Quién debe realizar el proceso?: un usuario

¿Cuál es el proceso que realiza …?

**Diseño**:

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA**: Usuario |
| **VARIABLES**  numeroA, numeroB: Entero // almacenan los números  suma, resta, multiplicación, división: Real // almacena el resultado |
| **NOMBRE ALGORITMO**: sumar\_restar\_multiplicar\_dividir\_numeros  **PROCESO DEL ALGORITMO**   * + - 1. *Leer* numeroA       2. *Leer* numeroB       3. suma ← numeroA + numeroB // suma y asigna el resultado       4. resta ← numeroA – numeroB // resta y asigna el resultado       5. multiplicacion ← numeroA \* numeroB       6. division ← numeroA / numeroB       7. *Mostrar* suma       8. *Mostrar* resta       9. *Mostrar* multiplicacion       10. *Mostrar* division |

Ejercicio 16: Necesitamos convertir una temperatura Fahrenheit en grados Celsius. Si no conoce la forma en la que se realiza esta conversión, debería investigarlo; para eso sirve la etapa de análisis. Pero como somos buenos, daremos una ayuda

**Definición del Problema**: Convertir una temperatura en Fahrenheit a grados Celsius.

**Análisis**:

* Datos de Entrada:

temperaturaFahrenheit: Real

* Datos de Salida:

temperaturaCelsius: Real

Proceso:

¿Quién debe realizar el proceso?:

Un químico

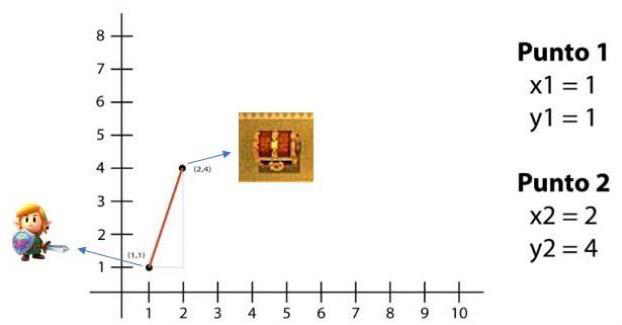
¿Cuál es el proceso que realiza el quimico?

temperaturaCelsius = (temperaturaFahrenheit – 32) / 1,8

**Diseño**:

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA**: |
| **VARIABLES**  temperaturaFahrenheit: Real // almacenan grados Fahrenheit  temperaturaCelsius: Real // almacena el resultado en grados Celsius |
| **NOMBRE ALGORITMO**: Convertir\_Grados\_Fahrenheit\_A\_Grados\_Celsius  **PROCESO DEL ALGORITMO**   * + - 1. *Leer* temperaturaFahrenheit       2. temperaturaCelsius ← (temperaturaFahrenheit – 32) / 1.8       3. *Mostrar* temperaturaCelsius |

Ejercicio 17: Si queremos representar personajes o power ups (premios) en la pantalla debemos primero ubicarlos en alguna posición dentro de la pantalla. Imagine que está en un juego donde un power up desaparece porque el personaje se acerca a una distancia de x unidades, sin importar por donde se acerque. Por tanto, para que desaparezca, en primer lugar, hay que determinar esa distancia. La forma de representar la posición de un objeto en la pantalla es a través de las coordenadas de un punto. Suponga que la posición de Link está representada por la coordenada (𝑥1, 𝑦1) , mientras que las de la caja de tesoro se halla en la posición (𝑥2, 𝑦2). Si observa con detenimiento se observa la conformación de un triángulo rectángulo, por lo que es posible aplicar Pitágoras para obtener la distancia



Para esto debe calcular el tamaño de los catetos y luego aplicar el teorema. Halle la distancia entre ambos objetos. Cuando programe, represente a lLnk con un Circulo, y al tesoro con un cuadrado. Además, mueva a Link mediante el mouse.

**Definición del Problema**: Calcular la distancia entre el personaje y el tesoro mediante Pitágoras

**Análisis**:

* Datos de Entrada: posición vectorial del personaje

posición vectorial del tesoro

* Datos de Salida: distancia entre el personaje y el tesoro
* Proceso:

¿Quién debe realizar el proceso?: el tesoro (objeto obtenible) es quien determina la su distancia y aparición ante el jugador

¿Cuál es el proceso que realiza? Gráficamente, trazamos un triángulo rectángulo entre la posición del jugador y el objeto obtenible. Luego, mediante el teorema de Pitágoras, podemos calcular la distancia entre ambos.

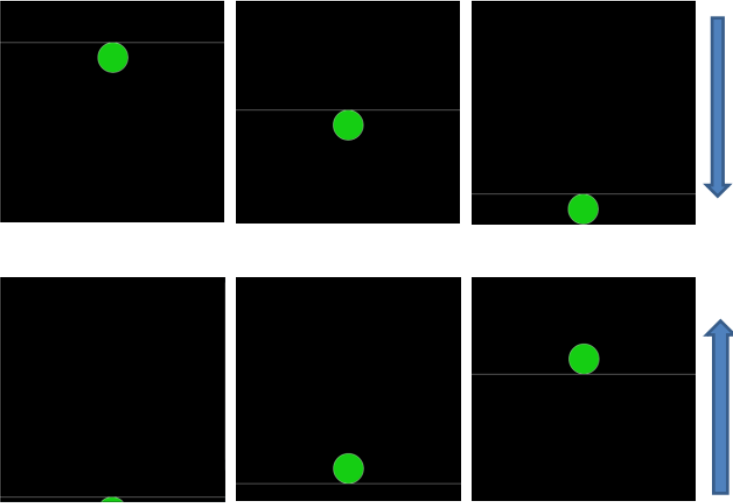
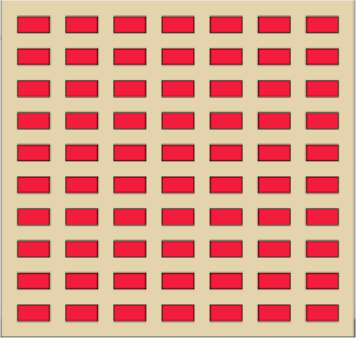
(distancia^2 = (posición del personaje en X – posición del tesoro en X)^2 + (posición del personaje en y – posición del tesoro en Y)^2)

**Diseño**:

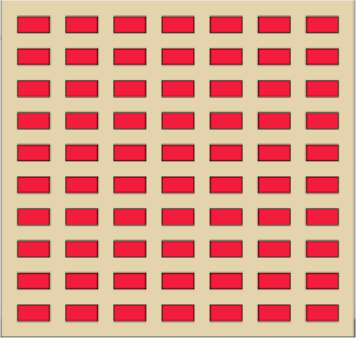
|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA**: Objeto Obtenible |
| **VARIABLES**  x1, x2, y1, y2: Entero // almacenan los números a sumar  distancia: Real // almacena el resultado |
| **NOMBRE ALGORITMO**: calcular\_distancia\_personaje\_tesoro  **PROCESO DEL ALGORITMO**   * + - 1. *Leer* x1       2. *Leer* y1       3. *Leer* x2       4. *Leer* y2       5. distancia ← ((x2-x1) ^2 + (y2-y1) ^2) ^(1/2)       6. *Mostrar* distancia |

Ejercicio 18: Desarrolle el análisis y diseño de un algoritmo que permita obtener las raíces de una ecuación de segundo grado. Además, utilice la estructura según para el análisis de la discriminante de la ecuación cuadrática. Obviamente codifique en Processing.

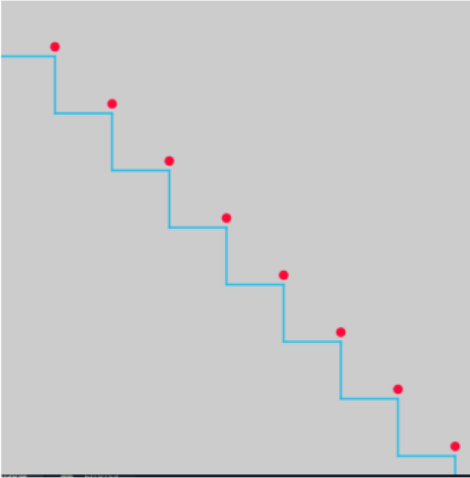
Ejercicio 19: Declare las variables necesarias para dibujar una línea que se dibuja desde las coordenadas iniciales del lienzo y se extiende por todo el ancho. Sobre el punto medio de la línea y a una distancia de 40px (en sentido vertical desde la línea) dibuje una elipse que tenga como ancho 80px y de alto 80px. Dentro de la función draw(), actualice las variables necesarias para que la línea desde su inicio se mueva en dirección hacia abajo arrastrando la elipse. Mantenga en cero el valor para background(). Cuando la línea supere la posición de la altura del lienzo, debe invertir su sentido, es decir dirigirse hacia arriba arrastrando la elipse. Cuando la línea alcance nuevamente el valor 0 para su posición en y, el desplazamiento debe ser hacia abajo y así sucesivamente. El lienzo debería verse como en las siguientes figuras



Ejercicio 20: Dibuje en toda la extensión del lienzo de (440, 420) rectángulos de idénticas medidas (40 ancho y 20 de alto) y que mantengan una distancia de 20 pixeles entre ellos tanto horizontal como verticalmente. Utilice la estructura de control repetitiva for. El lienzo debería verse así:

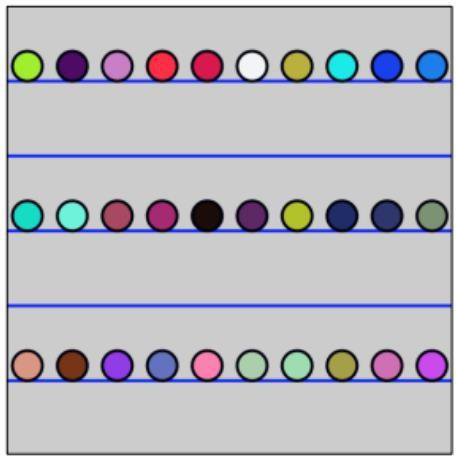


Ejercicio 21: Utilizando la estructura de control repetitiva while() dibuje la siguiente imagen utilizando líneas que forman escalones y sobre cada borde de escalón se dibuje un punto de color rojo



El tamaño del lienzo es size(500,500). La estructura while() se ejecuta dentro de la función setup(). La condición es que solo se dibuje dentro del lienzo. Utilice variables que puedan ayudar a la construcción del dibujo, por ej: x, y, anchoEscalon, altoEscalon, etc.

Ejercicio 22: Utilizando la estructura de control repetitiva do-while. Replique la siguiente imagen



La imagen debe ser construida desde la función setup(). Defina el tamaño del lienzo en size(600,600), verticalmente se divide el lienzo en franjas de igual medida, se deben dibujar los círculos sobre cada línea de por medio es decir en la línea 1 se dibujan círculos con distanciamiento, en la línea 2 no se dibuja y así sucesivamente. Las líneas tienen un color fijo, los círculos asumen colores aleatorios.

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)