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

Trabajo Práctico/Actividad

N° 1

Apellido y Nombre – LU / Sosa Juan Manuel / TUV000637

*Profesores:*

*Mg. Ing. Ariel Alejandro Vega*

**FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS**

* 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 \* 1 / 2) + (3 / 1 / 25 / 8)

1 + 24 / 25

0,96

* 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

Valor de a = 1

Valor de b = 2

Valor de c = 3

b ^ 2 – 4 \* a \* c

2 ^ 2 – 4 \* 1 \* 3

4 – 12

-8

* Expresión algebraica b2­­­ – 4.a.c

1. 3 \* X ^ 4 – 5 \* X ^ 3 + X 12 – 17

Valor de X = 5

3 \* X ^ 4 – 5 \* X ^ 3 + X 12 – 17

3 \* 5 ^ 4 – 5 \* 5 ^ 3 + 5 \* 12 – 17

3 \* 625 – 5 \* 125 + 60 - 17

1875 - 625 + 43

1293

* Expresión algebraica 3. X4 – 5. X3 + X.12 -17

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

1. (b + d) / (c + 4)

Valor de b = 7

Valor de c = 8

Valor de d = 5

(b + d) / (c + 4)

(7 + 5) / (8 + 4)

1

* Expresión algebraica

1. (x ^ 2 + y ^ 2) ^ (1 / 2)

Valor de x = 2

Valor de y = 1

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente(x ^ 2 + y ^ 2) ^ (1 / 2)

(2 ^ 2 + 1 ^ 2) ^ (1 / 2)

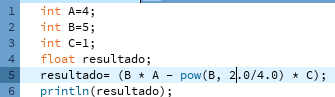
(4+1)^(1/2)

5^(1/2)

2.5

* Expresión algebraica (x2+y2)1/2
* 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:

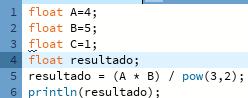
1. B \* A – B ^ 2 / 4 \* C

5 \* 4 – 5 ^ 2 / 4 \* 1

20 – 50,5\*1

20 – 2,23

17,76

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

(4 \* 5) / 3 ^ 2

20/9

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

Imagen que contiene Tabla

Descripción generada automáticamente(((5+1) /2\*4+10) \*3\*5)-6

((6/2\*4+10) \*3\*5)-6

((3\*4+10) \*3\*5)-6

((12+10) \*3\*5)-6

(22\*3\*5)-6

330-6

324

Ejercicio 6:

Para x=3, y=4; z=1, evaluar el resultado de

R1 = y+z

Texto

Descripción generada automáticamenteR2 = x >= R1

R1= y + z

5

R2= x > =R1

R2 = 3 >= 5

Ejercicio 7:

Para contador1=3, contador2=4, evaluar el resultado de

R1 = ++contador1

R2 = contador1 < contador2

Interfaz de usuario gráfica, Texto, Aplicación, Tabla

Descripción generada automáticamente con confianza mediaR1=++3

R=4<contador2

Ejercicio 8:

Para a=31, b=-1; x=3, y=2,

evaluar el resultado de

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamentea+b-1 < x\*y

a+b-1 < x\*y

31 - 1 - 1 < 3 \* 2

29 < 6

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteEjercicio 9:

Para x=6 y=8

evaluar el resultado de

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

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

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteEjercicio 10:

Para i=22,j=3,

evaluar el resultado de !((i>4) || !(j<=6))

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

Ejercicio 11:

Calendario

Descripción generada automáticamente con confianza bajaPara a=34, b=12,c=8,

evaluar el resultado de

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

!(43+12==8) || (8!=0)CC(12-8>=19)

True|| True CC False

Tue || False

True

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

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.

Análisis

|  |
| --- |
| Datos de Entrada: nombre\_tipeado  Datos de Salida: mensaje\_saludar  ¿Quién debe realizar el proceso? El algoritmo  ¿Cuál es el proceso que resuelve? Ingresar un nombre y crea un saludo para esa persona |

Diseño

|  |
| --- |
| Entidad que resuelve el problema: Algoritmo  Variables:  nombre\_tipeado: String  mensaje\_saludar: String  Nombre algoritmo: nombre\_saludar   1. Inicio 2. Leer nombre\_tipeado 3. Mostrar mensaje\_saludar 4. fin |

Ejercicio 13:

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  Datos de salida: área, perímetro  ¿Quién debe realizar el proceso?  ¿Cuál es el proceso que resuelve? Calcula el área y perímetro. |

Diseño

|  |
| --- |
| Entidad que resuelve el problema: Persona  Variables   * base: float * altura: float * perímetro: float * area: float * PeriArea: String   Nombre del Algoritmo: calcular\_perimetroarea\_rect  Proceso:   1. Inicio 2. Leer base 3. Leer altura 4. Calcular perímetro 5. Calcular área 6. Mostrar PeriArea 7. Fin |

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

Análisis

|  |
| --- |
|  |

|  |
| --- |
| Datos de entrada: cateto1,cateto2  Datos de salida: hipotenusa  ¿Quién debe realizar el proceso?  ¿Cuál es el proceso que resuelve? Calcula la hipotenusa de un triangulo usando la formula correcta utilizando los datos de los catetos. |

Diseño

|  |
| --- |
| Entidad que resuelve el problema: Persona  Variables   * cateto1:it * cateto2: int * hipotenusa: float   Nombre del Algoritmo: calcular\_hipotenusa  Proceso:   1. Inicio 2. Leer cateto1 3. Leer cateto2 4. Calcular hipotenusa 5. Mostrar hipotenusa 6. Fin |

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

Análisis

|  |
| --- |
| Datos de entrada: numero1, numero2  Datos de salida: suma, resta, multiplicación, división  ¿Quién debe realizar el proceso? Algoritmo  ¿Cuál es el proceso que resuelve? Dados losnúmeros permita calcular la suma, resta, multiplicación y división entre estos |

Diseño

|  |
| --- |
| Entidad que resuelve el problema: Algoritmo  Variables   * numero2: float * numero1: float * suma: float * resta: float * multiplicación: float * division: float   Nombre del Algoritmo: calcular\_operaciones  Proceso:   1. Inicio 2. Leer numero1 3. Leer unmero2 4. Calcular suma 5. Mostrar suma 6. Calcular resta 7. Mostrar resta 8. Calcular multiplicación 9. Mostrar multiplicación 10. Calcular división 11. Mostrar división 12. Fin |

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.

Análisis

|  |
| --- |
| Datos de entrada: Temperatura en Fahrenheit  Datos de salida: Temperatura en Celsius  ¿Quién debe realizar el proceso? Algoritmo  ¿Cuál es el proceso que resuelve? Resuelve la conversión de grados Fahrenheit a Celsius. |

Diseño

|  |
| --- |
| Entidad que resuelve el problema: Algoritmo  Variables   * Temptcelsius:float * tempfareheit: float   Nombre del Algoritmo: calcular\_celsius  Proceso:   1. Inicio 2. Leer tempfareheit 3. Calcular tempcelsius 4. Mostar temptcelsius 5. Fin |

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.

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.

Análisis

|  |
| --- |
| Datos de entrada: coeficientes de la ecuación cuadrática  Datos de salida: Raíces de la ecuacion  ¿Quién debe realizar el proceso?  ¿Cuál es el proceso que resuelve? Calcular el discriminante de la ecuación cuadrática |

Diseño

|  |
| --- |
| Entidad que resuelve el problema: Algoritmo  Variables   * a: float * b: float * c: float * discrimiante: float * x1: float * x2: float * x: float * raices: String * raizdoble: String   Nombre del Algoritmo: discriminante\_ecuacionnes\_cuadraticas  Proceso:   1. Inicio 2. Leer a 3. Leer b 4. Leer c 5. Calcular discriminante 6. Si discriminante >0 entonces x1 y x2 7. Mostrar raíces 8. si\_no si discriminante==0 entonces x 9. Mostrar raizdoble 10. si\_no 11. Mostrar “no hay raíces reales” 12. Fin |

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:

Análisis:

|  |
| --- |
| * Datos de entrada   - coordenadasRect: coordenadas cartesianas  - ancho, alto, distanciaEntreRect: Entero   * Datos de Salida   - Los Rectángulos dibujados   * Proceso   - Dibujar rectángulos |

Diseño

|  |
| --- |
| Entidad: Lienzo  Variables:   * coordenadasRect: Coordenadas * anchorect: int * altorect: int * distEtreRect: int   Algoritmo   1. Inicio 2. Alto.Lienzo 440 3. Ancho.Lienzo 420 4. Distancia entre rectángulo 20 5. anchorec 40 6. altorec 20 7. Para x: coodernadasRect.x hasta anchoLienzo (ancho+distEntreRect) 8. Para y: coordenadasRect.y hasta altoLienzo (alto+distEntreRect) 9. Dibujar Rectángulos 10. Fin. |

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.

Análisis

|  |
| --- |
| Datos de entrada: punto1,punto2,punto3,punto4,distancia  Datos de salida: Escalones con puntos de colores  ¿Quién debe realizar el proceso? El programa processing  ¿Cuál es el proceso que resuelve? Dibujar escalones con puntos rojos mediante el while() |

Diseño

|  |
| --- |
| Entidad que resuelve: Algoritmo  Variables:   * punto1,punto2,punto3,punto4: int * distancia: int   Nombre de Algoritmo: while\_puntos\_escalones  Proceso:   1. Inicio 2. anchoLienzo 500 3. altoLienzo 500 4. distancia 60 5. mientras punto1.y sea menor o igual que ancho Lienzo 6. Dibujar linea horizotal en punto1.x, punto1.y, punto2.x, punto2.y 7. Dibujar línea vertical en punto2.x, punto2.y, punto3.x, punto3.y 8. Dibujar circulo en punto4.x, punto4.y 9. Fin\_mientras |