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

TRABAJO PRACTICO N°1

Operadores – Metodología de Programación

**RASKOVSKY DAVID ADRIAN**

L.U. TUV000629

**FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS**

*Profesores:*

*Mg. Ing. Ariel Alejandro Vega*

*Año 2024*

Índice

**Pag.2.1 – Resolver:………**

**Pag.2.1.1 Ejercicio 1: Evaluar (obtener resultado) la siguiente expresión para A = 2 y B = 5**

**Pag.2.1.2 Ejercicio 2: Evaluar la siguiente expresión**

**Pag.2.1.3. 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**

**Pag.2.1.4 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.**

**Pag.3.1.1.1 a)b ^ 2 – 4 \* a \* c**

**Pag.3.1.1.2 b) 3 \* X ^ 4 – 5 \* X ^ 3 + X 12 – 17**

**Pag.3.1.1.3 c) *d / (c + 4)***

**Pag.3.1.1.4 d) (x ^ 2 + y ^ 2) ^ (1 / 2)**

**Pag.3.41 - ………… Ejercicio**

**Pag.3.5 - ………… Ejercicio**

**Pag.3.6 - ………… Ejercicio**

**Pag.3.7 - ………… Ejercicio**

**Pag.3.8 - ………… Ejercicio**

**Pag.3.9 - ………… Ejercicio**

**Pag.3.10 - ………… Ejercicio**

**Pag.3.11- ………… Ejercicio**

***TRABAJO PRACTICO TALLER***

**Pag.. - …………………….. Ejercicio 1: Dibujar sobre un lienzo de tamaño (400,400) 4 puntos**

**Pag.. - ………………………………….. Ejercicio punto 2 Dibjar líneas de colores en un lienzo.**

**Pag.. - ………………………………….. Ejercicio punto 3 Dibjar rectángulos con for en un lienzo.**

**Pag.. - ………………………………….. Ejercicio punto 4 Dibjar Circulos de colores en línea media.**

**Pag..1 - ………………………………….. Ejercicio punto 5 Dibjar puntos cardinales en lienzo.**

**Pag..1 - ………………………………….. Ejercicio punto 6 Dibjar puntos de colores y posición aleatoria en lienzo.**

**Pag..1 - ………………………………….. Ejercicio punto 7 Dibjar con primitivas un tractor a colores**

**Pag..1 - ………………………………….. Ejercicio punto 7 Dibjar**

***2.1-***

|  |  |
| --- | --- |
| ***2.1.1 Ejercicio 1:*** Evaluar (obtener resultado) la siguiente expresión para  A = 2 y B = 5 => 3\* A - 4 \* B / A ^ 2 = 3\*2 - 4\*5 /4  = 6 – 4\*5 / 4  = 6 – 20 / 4 =  = 6 – 5  = 1 | |
| Resolución necesaria en Word:  (3\*A)-(4\*B/(A^2))  6-(4\*B/4)  6 - 5 =  **1** | Captura de Processing |

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

4 / 2 \* 3 / 6 + 6 / 2 / 1 / 5 ^ 2 / 4 \* 2 => por asociatividad nos queda

|  |  |
| --- | --- |
| Resolución necesaria en Word:  4/2\*3/6+6/2/1/5 ^2/4\*2 = >  2\*3/6+6/2/1/25/4\*2 =>  6/6+6/2/1/25/4\*2 =>  1+6/2/1/25/4\*2 =>  1+3/1/25/4\*2 =>  1+3/25/4\*2 =>  1+0.12/4\*2 => **(ojo)**  1+0.03\*2 =>  1+0.06 =>  = **1.06**  1+0.12/4\*2 => **(ojo)**  1+0.12/8 =>  1+0.015 =>  = **1.015 (mal)** | Captura de Processing |

***2.1.3 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.

**𝑏2 − 4. 𝑎. 𝑐** = b^2 - 4\*a\*c

***2.1.4 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.**

***3.1.1.1***

|  |  |  |
| --- | --- | --- |
| 1. ***b ^ 2 – 4 \* a \* c***   **a=2, b= 6, c= 4** | (b\*b) - 4\*a\*c =>  (36) – 4\*a\*c =>  36 – 8\*c => 36 - 32 =>  36 - 32 = **4** | **B2-4ac = (b.b) - ((4a)c)** |
| ***.3.1.1.2*** |  |  |
| 1. **3 \* X ^ 4 – 5 \* X ^ 3 + X 12 – 17**   **X= 2** | 3\*(16)-5\*(2**^**3)+X12-17 =)>  3\*16 – 5 \* 8 + X12-17 =>  48 - 5 \* 8 + X12 – 17 =>  48 – 40 + X \* 12 – 17 =>  48 – 40 + 24 – 17 =>  8 + 24 – 17 =>   1. - 17 = **15** | **= 3(X4) – 5( X3) + X 12 – 17** |
| ***3.1.1.3*** |  |  |
| 1. ***+ d / (c + 4) =>***   ***b= 3, c= 4, d= 2*** | (3 + 2) / (4 + 4) =>  (5) / (8) =>  5 / 8 = **0.625** | = **(b + d)**  **(c + 4)** |
| ***3.1.1.4*** |  |  |
| 1. **(x ^ 2 + y ^ 2) ^ (1 / 2)**   **x= 3, y= 4,** | (9 + y ^ 2) ^ (1 / 2) =>  (9+16) ^ (1/2) =>  25 ^ (1/2) =>  25 ^ 0.5 = **5** | **(x2 + y2 )(1/2) =(x2 + y2 )(0.5)** |

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

𝑏2 − 4. 𝑎. 𝑐

***3.1.2-***

***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:

|  |  |  |
| --- | --- | --- |
| **B \* A – B ^ 2 / 4 \* C** | B.A-B^2/4\*C =>  B.A - (25) /4\*C  20 – 25 / 4\*C  20- 6.25 \*1  20- 6.25 = **13.75** | **\_ B . A – B2\_\_\_\_\_\_\_\_ = 13,75**  **4 . C** |
| 1. **(A \* B) / 3 ^ 2** | (AB)/ 3 ^ 2 =>  (A \* B)/ 9 =>  20 / 9 = **2.222** | **(A . B)**  **=**  **3 2** |

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

***3.2***

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

|  |  |
| --- | --- |
| **R1 = y+z**  **R2 = x >= R1** | R1=7 =>  => R2 = 3 >= 7 => R2 = **False** |

***3.3***

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

|  |  |
| --- | --- |
| R1 = ++contador1 =>  R2 = contador1 < contador2 | => R1= 4 , R2= 4 =>  => R2= 4 < 4 => R2= **False** |

***3.4***

***Ejercicio 8:*** Para a=31, b=-1; x=3, y=2, evaluar el resultado de

|  |  |
| --- | --- |
| **a+b-1 < x\*y** => | a + b – 1 < 18 => 31 -1 - 1 < 18 => 30 - 1 <18 => 29 < 18 =>  **False** |

***3.5***

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

!(x<5)CC !(y>=7) => !(6 < 5)CC !(8 >= 7) => !(F)CC !(T) => T CC F = F

***3.6***

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

***3.7***

!((i>4) || !(j<=6)) => !((22>4) || !(j<=6)) => !((T) || !(3<=6)) => !(T || ! T) => !(T || !T) => !(T || F) => !(T) = F

***3.8***

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

!(a+b==c) || (c!=0)CC(b-c>=19) => !(34+12==8) || (8!=0)CC(12 – 8 >= 19) => !(46==8) || (8!=0)CC(4>=19) =>

=> !(F) || (T) CC (F) => T || T CC F => T CC F = F

**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.

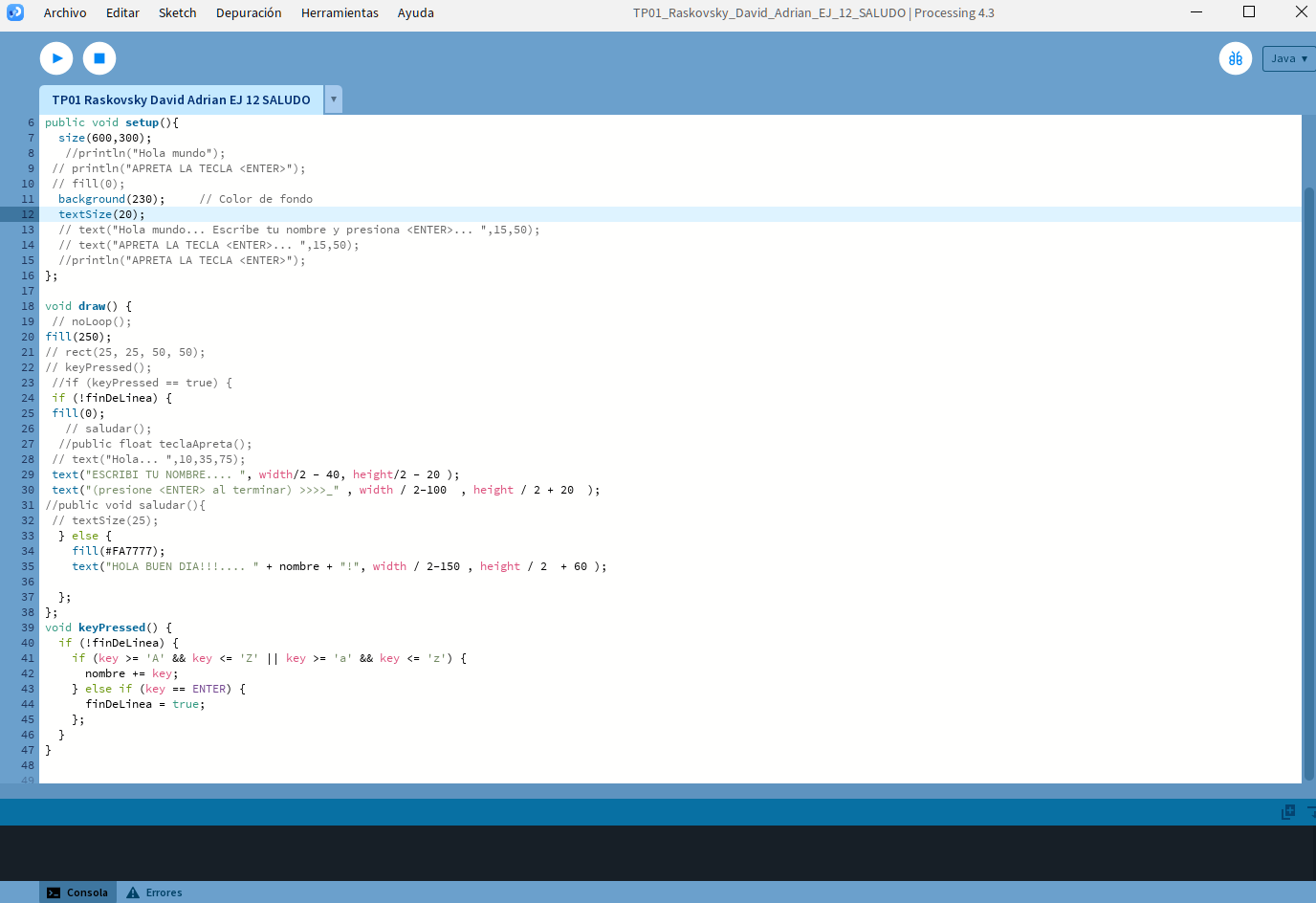
***DESARROLLO DEL PUNTO***

|  |
| --- |
| ***DESCRIPCION DEL PORBLEMA:*** Escribir en pantalla el nombre pedido al usuario |
| ***Analasis***: |
| ***DATOS DE ENTRADA*** |
| String nombre = ""; // Variable para almacenar el nombre del usuario  boolean finDeLinea= false // variable que controla el final del nombre |
| ***DATOS DE SALIDA*** |
| Escribir(“Hola mundo, escriba su nombre , presione <ENTER AL TERMINAR> ” + nombre ) |
| ***PORCESO*** |
| Procedure  si (!finDeLinea) {  si (key >= 'A' && key <= 'Z' || key >= 'a' && key <= 'z') {  nombre += key;  } sino {  si (key == ENTER) {  finDeLinea = true;}  };  fin Procedure  fin |

***DISEÑO***

|  |
| --- |
| **ENTIDAD**: LIENZO |
| **VARIABLES:** String nombre = ""; // Variable para almacenar el nombre del usuario  boolean finDeLinea= false // variable que controla el final del nombre |
|  |
| **ALGORITMO NOMBRE**: Hola mundo, escriba su nombre  **ALGORITMO:**  **inicio**  **String nombre = ""; // Variable para almacenar el nombre del usuario**  **boolean finDeLinea= false;**  **if (!finDeLinea) {**  **text("ESCRIBI TU NOMBRE.... ");**  **text("(presione <ENTER> al terminar) >>>>\_");**  **} else {**  **text("HOLA BUEN DIA!!!.... " + nombre + "!");**  **proceso keyPressed()**  **if (!finDeLinea) {**  **if (key >= 'A' && key <= 'Z' || key >= 'a' && key <= 'z') {**  **nombre += key;**  **} else if (key == ENTER) {**  **finDeLinea = true; };**  **}**  **Fin proceso**  Fin |

***CODIGO:***



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

***DESARROLLO DEL PUNTO***

|  |
| --- |
| ***DESCRIPCION DEL PORBLEMA:*** Escribir en pantalla el nombre pedido al usuario |
| ***Analasis***: |
| ***DATOS DE ENTRADA*** |
| String nombre = ""; // Variable para almacenar el nombre del usuario  boolean finDeLinea= false // variable que controla el final del nombre  areaRectangulo, perimetrRectangulo : float |
| ***DATOS DE SALIDA*** |
| //Escribe en pantalla el Area y el Perimetro del rectángulo  Escribir(“EL VALOR DEL AREA ES = “+ areaRectangulo )  Escribir(“EL VALOR DEL AREA ES = “ + perometrRectangulo ) |
| ***PORCESO*** |
| Procedure  Escribir(“INGRESE LA BASE DEL RECTANGULO, … Y ´PRESIONE TECLA <ENTER>)  //Leer(baseRectangulo)  si (!finDeLinea) {  si (key >= 'A' && key <= 'Z' || key >= 'a' && key <= 'z') {  Escribir(“INGRESE UN NUMERO”)  } sino {  baseRectangulo += key;  si (key == ENTER) {  finDeLinea = true;}  };  Escribir(“INGRESE LA ALTURA DEL RECTANGULO,.. Y ´PRESIONE TECLA <ENTER>)  //Lee la altura  si (!finDeLinea) {  si (key >= 'A' && key <= 'Z' || key >= 'a' && key <= 'z') {  Escribir(“INGRESE UN NUMERO”)  } sino {  alturaRectangulo += key; // acumulador  si (key == ENTER) {  finDeLinea = true;}  };  areaRectangulo = baseRectangulo \* alturaRectangulo  perimetrRectangulo = baseRectangulo + baseRectangulo + alturaRectangulo+ alturaRectangulo  Escribir(“El Area del rectángulo es:” + areaRectangulo )  Escribir(“El Area del rectángulo es:” + areaRectangulo )  fin Procedure  fin |

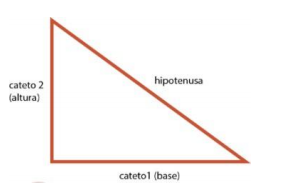
***DISEÑO***

|  |
| --- |
| **ENTIDAD**: LIENZO |
| **VARIABLES:** String nombre = ""; // Variable para almacenar el nombre del usuario  boolean finDeLinea= false // variable que controla el final del nombre |
|  |
| **ALGORITMO NOMBRE**: Hola mundo, escriba su nombre  **ALGORITMO:**  **inicio**  **String nombre = ""; // Variable para almacenar el nombre del usuario**  **boolean finDeLinea= false;**  **if (!finDeLinea) {**  **text("ESCRIBI TU NOMBRE.... ");**  **text("(presione <ENTER> al terminar) >>>>\_");**  **} else {**  **text("HOLA BUEN DIA!!!.... " + nombre + "!");**  **proceso keyPressed()**  **if (!finDeLinea) {**  **if (key >= 'A' && key <= 'Z' || key >= 'a' && key <= 'z') {**  **nombre += key;**  **} else if (key == ENTER) {**  **finDeLinea = true; };**  **}**  **Fin proceso**  Fin |

***Ejercicio 14:***

asumir que su gran amigo son las matemáticas. Obtenga la hipotenusa de un triángulo

rectángulo conociendo sus catetos



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.

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



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

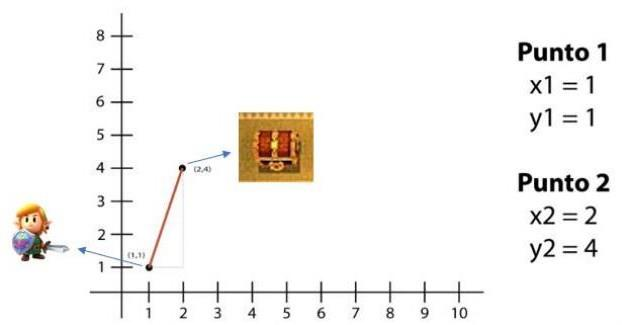
x unidades, sin importar por donde se acerque. Por tanto, para que desaparezca, en primer

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 (?1 , ?1 ) , mientras que las de la caja de tesoro se halla

en la posición (?2 , ?2 ). Si observa con detenimiento se observa la conformación de un

triángulo rectángulo, por lo que es posible aplicar Pitágoras para obtener la 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

***EJERCICI 20 - Ejercicio 3 – Taller :*** 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 utilizando estructuras iterativas para que se vea de la siguiente manera:

DESCRIPCION DEL PORBLEMA; Dibujar en processing rectángulos con estructura iterativa .

***ANALASIS:***

***DATOS DE ENTRADA***

Tamaño del lienzo (440, 420)

coordenadasRect: Coordenadas cartesianas

anchoRect, altoRect, espacioEntreRect: Entero

***DATOS DE SALIDA***

Dibuja los rectángulos en lienzo.

***PORCESO***

Dibjar los Rectangulos

***DISEÑO***

|  |
| --- |
| **ENTIDAD**: LIENZO |
| **VARIABLES:** |
| coordenadasRect: coordenadas  anchoRect, altoRect, espacioEntreRect: Entero  anchoLienzo, altoLienzo: Entero |
| Nombre algoritmo: dibujar\_rectangulos |
| **ALGORITMO:**  Inicio  anchoLienzo <- 400;  altoLienzo <- 420;  espacioEntreRect <- 20  anchoRect <- 40  altoRect <- 20  // USAMOS DOS FOR PARA MOSTRA LA MATRIZ DE RECTANGULOS  para y<- coordenadasRect.y hasta altoLienzo con paso (anchoRect+ espacioEntreRect) // **y= y + anchoRect + espacioEntreRect**  hacer  Para x <- coordenadasRect.x hasta anchoLienzo con paso (anchoRect+ espacioEntreRect)  // x= x + anchoRect + espacioEntreRect  hacer  //Dibujar un rectángulo en (espacioEntreRect, espacioEntreRect);  Dibujar un rectangulo en (x, y) con dimenciones anchoRect altoRect;  Fin para  Fin para  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

Necesito los puntos y coordenadas para las líneas rectas:

Objetos Punto1, Punto2, Punto3 , variable : distanciaEscalon

linea1(x,y,z,w)=> lionea1(Punto1.x, Punto1.y, Punto2.x, Punto2.y) =>

Punto1(X,Y)=>A1(0, distanciaEscalon) y Punto2(distanciaEscalon, distanciaEscalon) =>

linea1(x,y,z,w)=>linea1(0, distanciaEscalon, distanciaEscalon, distanciaEscalon) =>

**para la línea del horizontal del escalon**

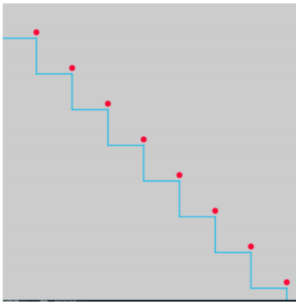
linea2(x,y,z,w)=> lionea2(Punto2.x, Punto2.y, Punto3.x, Punto3.y) =>

Punto1(X,Y)=>A1(0, distanciaEscalon) y Punto2(distanciaEscalon, distanciaEscalon) =>

linea1(x,y,z,w)=>linea1(0, distanciaEscalon, distanciaEscalon, distanciaEscalon) =>

**para la línea del vertical del escalon**

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

***DESARROLLO DEL PUNTO***

***DESCRIPCION DEL PORBLEMA***; Dibujar en Processing una línea celeste de 8 escalones cun un punto rojo cada escalon y en diagonal de izquierda a derecha.

***ANALASIS***:

DATOS DE ENTRADA

Punto1, punto2, punto3, punto4 : cooredenadas cartecianas en 2D;

DistanciDelEscalon: emtero

DATOS DE SALIDA

El dibujo de la línea horizontal.

El dibujo de la línea vertical.

El dibujo del punto rojo.

PORCESO

Dibuja una línea horizontal entre el punto1 y punto2, con distancia = distanciDelEscalon

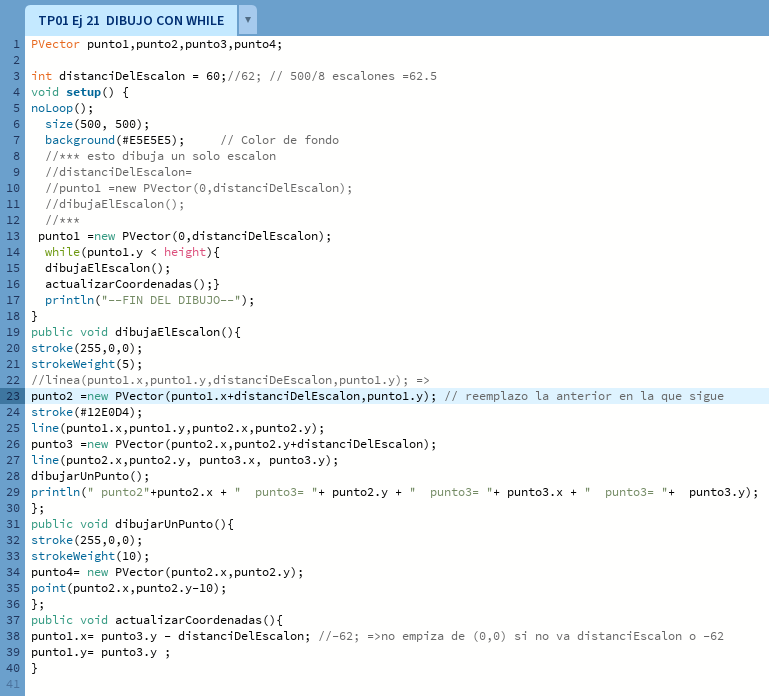
Dibuja una línea vertical entre el punto2 y punto3, con distancia = distanciDelEscalon

Dibuja un punto rojo en la siguiente posición: x = posición en x del punot2 entre el punto1, y= posición en y del punto2 – 5 unidades

Repetir desde el principio hasta que las coordenadas en y del punto1 sea mayor que el alto del lienzo.

***DISEÑO***

|  |
| --- |
| **ENTIDAD**: escalón |
| **VARIABLES:** |
| Punto1, punto2, punto3, punto4 : cooredenadas cartecianas en 2D;  DistanciDelEscalon: entero |
| ***NOMBRE ALGORITMO***: dibujar\_escalon.  ***ALGORITMO***  Inicio   * Dibujar una línea horizontal entre el punto1 y punto2 con distancia = distanciDelEscalon * Dibuja una línea vertical entre el punto2 y punto3 con distancia a = distanciDelEscalon * Dibuja un circulo rojo en la siguiente posición: x = posición en x de punot2 entre el punto1, y= posición en y del punto2 – 5 unidades * Actualizar las coordenadas de punta con los del punto3 * Repetir desde el principio hasta que las coordenadas en y del punto1 sea mayor que el alto del lienzo.   Fin |
| Nombre algoritmo: Dibujo círculo rojo  ***ALGORITMO***  Inicio   * Dibujar un punto en la siguiente posición: x= posicion en x de punto2, y= posición en y de punt2 – 10   Fin |

**CODIFICACION:**

***Punto 1.1*** *– Taller* ***:*** Enunciado del punto

Ejercicio 1: Dibujar sobre un lienzo de tamaño (400,400) 4 puntos, los cuales deben estar ubicados en las 4 esquinas del lienzo dejando un margen de 20 pixelestanto en las coordenadas x como en las coordenadas y. Dibuje un punto en el centro del lienzo, para ello utilice las constantes width (ancho) y height(alto) que representan el ancho y el alto del lienzo. Utilice las funciones point(x,y), strokeWeight(10).

Deberíamos ver algo como esto:



Luego agregar los puntos medios de cada lado del lienzo.

***DESARROLLO DEL PUNTO***

|  |
| --- |
| ***DESCRIPCION DEL PORBLEMA***; Dibujar en processing los puntos coordenados |
| Analasis: |
| ***DATOS DE ENTRADA*** |
| puntoA, puntoB, puntoC, puntoD : cooredenadas |
| ***DATOS DE SALIDA*** |
| Dibuja los cuatro puntos. |
| ***PORCESO*** |
| //Dibuja el punto A en (distanciaAlBorde, distanciaAlBorde);  puntoA= new PVector (distanciaAlBorde, , distanciaAlBorde)  Point(puntoA.x ,puntoA y) |

***DISEÑO***

|  |
| --- |
| **ENTIDAD**: LIENZO |
| **VARIABLES:** |
| puntoA, puntoB, puntoC, puntoD, puntoE : coordenadas  anchoLienzo, altoLienzo, distanciaAlBorde: enteros |
| Nombre algoritmo: dibujar\_puntos  **ALGORITMO:**  Inicio  anchoLienzo <- 400;  altoLienzo <- 400;  Dibuje el punto A en (20,20);  Dibuje el punto B en (380,380);  Dibuje el punto C en (20,380);  Dibuje el punto D en (380,20);  Dibuje el punto E en (200,200); // PUNTO AL MEDIO  distanciaAlBorde <- 20;  Dibuje el punto A en (distanciaAlBorde, distanciaAlBorde);  Dibuje el punto D en (anchoLienzo - distanciaAlBorde, distanciaAlBorde);  Dibuje el punto C en (distanciaAlBorde, altoLienzo - 20);  Dibuje el punto B en (anchoLienzo - distanciaAlBorde, altoLienzo-20);  Dibuje el punto E en (anchoLienzo /2, altoLienzo /2); // PUNTO DEL MEDIO  Fin |

**CODIFICACION:**





**Ejercicio 2 - taller:**

Dibuje las líneas necesarias para dividir el lienzo de tamaño (400,400) como se muestra en la figura, utilice diferentes colores para cada línea. También dibuje un punto en el centro del lienzo. Utilice line(x1, y1, x2, y2), strokeWeight(4), stroke(x,x,x), point(x,y) El lienzo debería verse como en la siguiente figura:

Necesito los puntos y coordenadas para las líneas rectas:

A(X,Y)=>A1(0,0) y A2(ancho, alto) = A2(400,400) **para la línea amarilla**

B(X,Y)=>B1(0,alto/2) y B2(ancho, alto/2) **para la línea roja**

C(X,Y)=>C1(0,alto) y C2(ancho,0) **para la línea celeste**

D(X,Y)=D1(0,ancho/2) y D2(ancho/2, alto) **para la línea azul**



Desarrollo del punto

DESCRIPCION DEL PORBLEMA; Dibujar en processing las líneas diagonales y perpendiculares.

Analasis:

DATOS DE ENTRADA

puntoA, puntoB, puntoC, puntoD : cooredenadas

DATOS DE SALIDA

Dibuja los cuatro puntos.

PORCESO

DISEÑO

|  |
| --- |
| **ENTIDAD**: LIENZO |
| **VARIABLES:** |
| PuntoMedio, puntoA1, puntoA2, puntoB1, puntoB2, puntoC1, puntoC2, puntoD1, puntoD2: coordenadas;  lineaAmarilla, lineaRoja, lineaCeleste, lineaAzul: string;  anchoLienzo, altoLienzo: enteros; |
| Nombre algoritmo: dibujar\_lineas\_de\_colores |
| **ALGORITMO:**  Inicio  anchoLienzo <- 400;  altoLienzo <- 400;  // Dibuje el lineaAmarilla en (puntoA1, puntoA2);  // Dibuje el lineaRoja en (puntoB1, puntoB2);  // Dibuje el lineaCeleste en (puntoC1, puntoC2);  // Dibuje el lineaAzul en (puntoD1, puntoD2);  Dibuje el punto puntoMedio en (20,20);  Dibuje el lineaAmarilla en (0,0, anchoLienzo, altoLienzo); // **para la línea amarilla**  Dibuje el lineaRoja en (0, altoLienzo /2, altoLienzo, altoLienzo /2); // **para la línea roja**  Dibuje el lineaCeleste en (0, altoLienzo,0, anchoLienzo); // **para la línea celeste**  Dibuje el lineaAzul en (0, anchoLienzo/2, altoLienzo /2, altoLienzo); // **para la línea azul**  Dibuje el punto puntoMedio en (anchoLienzo/2, anchoLienzo /2); // **para EL PUNTO AL MEDIO**  Fin |

**CODIFICACION:**



DIRECCION EN GITUB:

<https://github.com/adrianRDA24/RASKOVSKYDavidAdrianTP01_Ejerecicio02.git>

G:\FACULTAD UNJU\FUNDAMENTOS - RECURSADA\GitHub\TP01\_RASKOVSKY\_DAVID\_ADRIAN\_\TP01\_RASKOVSKY\_DAVID\_ADRIAN\_

***CODIFICACION*:**

***Ejercicio 4:***

***Ejercicio 4:***

Dibuje una línea sobre el lienzo de (400,400) que divida horizontalmente el mismo. Luego dibuje un círculo, cuyo centro coincida con el punto central del lienzo, el alto y ancho del círculo debe ser de 60 px. El círculo central debe ser de color rojo tanto sus bordes como su relleno. Utilice line(), circle(), stroke(), strokeWeight(), fill() Luego dibuje un círculo a la izquierda (color amarillo) y a la derecha (color verde) con una distancia de 150 px desde el centro del primer círculo. Use para el color de cada círculo un Según. Debe quedar así:

***Desarrollo del punto 4***

***DESCRIPCION DEL PORBLEMA***; Dibujar en processing línea y 3 Círculos de colores al medio.

***ANALASIS:***

***DATOS DE ENTRADA***

lienzo (400,400).

circuloA, circuloB, circuloC, lineacentral : cooredenadas.

***DATOS DE SALIDA***

Dibujar una lineacentral(x,y,x’,y’).

Dibujar las 3 circulos de colores sobre la lineacentral.

circuloA, circuloB, circuloC

***PORCESO***

Dibujar una lineacentral(x,y,x’,y’).

Dibuje el circuloA (20,20); //rojo

Dibuje el circuloB (380,380); //azul

Dibuje el circuloC en (20,380); //amarillo

***DISEÑO***

|  |
| --- |
| **ENTIDAD**: LIENZO |
| **VARIABLES:** |
| circuloA, circuloB, circuloC, puntoD, lineacentral: coordenadas  anchoLienzo, altoLienzo: enteros |
| Nombre algoritmo: dibujarLineaCentraly3Circulos  **ALGORITMO:**  Inicio  anchoLienzo <- 400;  altoLienzo <- 400;  Dibujar una lineacentral(0, altoLienzo/2, anchoLienzo, altoLienzo/2);  Dibuje el circuloA (anchoLienzo/2, altoLienzo/2); //rojo  Dibuje el circuloB (altoLienzo/2, anchoLienzo/3); //azul  Dibuje el circuloC en (anchoLienzo/3, altoLienzo/2); //amarillo  Dibuje el punto E en (anchoLienzo /2, altoLienzo /2); // PUNTO DEL MEDIO  Fin |

***CODIFICACION:***



***Ejercicio 5:***

Dibuje los puntos cardinales N, S, E, O utilizando figuras como rectángulos y triángulos.

Utilice: rect(), triangle(), background(), text(), fill(), stroke();

***Desarrollo del punto 4***

***DESCRIPCION DEL PORBLEMA***; Dibujar en Processing línea y 3 Círculos de colores al medio.

***ANALASIS:***

***DATOS DE ENTRADA***

lienzo (400,400).

***DATOS DE SALIDA***

Dibujar una con figuras básicas en el lienzo los puntos cardinales

Dibujar las 4 triángulos y 2 rectángulos y los puntos cardinales

***PORCESO***

Dibujar en el lienzo:

text("NORTE");

text("OESTE");

text("ESTE");

text("SUR");

rect(width/2 -20,55,40,300); // 400,400

rect(55,width/2 -20,300,40);

triangle((width/2),0,(width/2-30),60, (width/2+30), 60);

triangle(0,(width/2),60,(width/2-30),60, (width/2+30));

triangle((width/2),(height),(width/2+30),height-60,(width/2-30),(height-60));

triangle((height),(width/2),height-60,(width/2+30),(height-60),(width/2-30));

***DISEÑO***

|  |
| --- |
| **ENTIDAD**: LIENZO |
| **VARIABLES:** |
| anchoLienzo, altoLienzo: enteros |
| Nombre algoritmo: dibujar\_Puntos\_Cardinales  **ALGORITMO:**  Inicio  anchoLienzo <- 400;  altoLienzo <- 400;  // Dibujar:  text("NORTE");  text("OESTE");  text("ESTE");  text("SUR");  rect(width/2 -20,55,40,300); // 400,400  rect(55,width/2 -20,300,40);  triangle((width/2),0,(width/2-30),60, (width/2+30), 60);  triangle(0,(width/2),60,(width/2-30),60, (width/2+30));  triangle((width/2),(height),(width/2+30),height-60,(width/2-30),(height-60));  triangle((height),(width/2),height-60,(width/2+30),(height-60),(width/2-30));  Fin |

***CODIFICACION:***

Ejercicio 6:

Dibuje sobre un lienzo de (400,400) puntos de diferentes colores y en diferentes posiciones. Utilice la función random() para generar las posiciones x e y de los diferentes puntos y cambiar los colores.

***Desarrollo del punto 6***

***DESCRIPCION DEL PORBLEMA***; Dibujar en Processing puntos de colores aleatorios y posicion con random.

***ANALASIS:***

***DATOS DE ENTRADA***

lienzo (400,400).

***DATOS DE SALIDA***

Dibujar puntos de colores y posición random en el lienzo

***PORCESO***

para (int y = 0; y < height; y++)

hacer

Para (int x = 0; x < width; x++)

hacer

float r = random(256);

float g = random(256);

float b = random(256);

stroke(r,g,b);

fill(r,g,b);

ellipse(random(x),random(y),random(15),random(15));

Fin para

Fin para

***DISEÑO:***

|  |
| --- |
| **ENTIDAD**: LIENZO |
| **VARIABLES:** |
| anchoLienzo, altoLienzo: enteros |
| Nombre algoritmo: dibujar\_Puntos\_Aleatorios\_de\_colores  **ALGORITMO:**  Inicio  anchoLienzo <- 400;  altoLienzo <- 400;  // Dibujar:  float r = random(256);  float g = random(256);  float b = random(256);  stroke(r,g,b);  fill(r,g,b);  ellipse(random(x), random(y), random(15), random(15));  Fin |
|  |

***CODIFICACION:***



***Ejercicio 7:*** Trate de replicar el siguiente dibujo:

***DESCRIPCION DEL PORBLEMA***; Dibujar en Processing puntos de colores aleatorios y posicion con random.

***ANALASIS:***

***DATOS DE ENTRADA***

lienzo (400,400).

***DATOS DE SALIDA***

Dibujar un tren de colores con primitivas en el lienzo

***PORCESO***

float X;

float Y;

Dibujar

X= width ;

Y= height;

float rectX= X/4; //rectangulo en X

float rectY= Y-2; //rectangulo en Y

rect(rectX, 0 ,rectY-100, Y); // cuadro del fondo

fill(#FF0303); //relleno rojo

rect(rectX+25,rectY-100,30,-100);

fill(#09D636);

rect(rectX+25,rectY-50,153,-100); //VERDE - rectX,rectX - 50);

fill(#2DD8ED);

rect(rectX+25+80,rectY-120,65,-120); // cabina ceñeste

ellipse(rectX+50+2,rectY-40,50,50);// rueda celeste

fill(#FF0303);

ellipse(rectX+150+2,rectY-40,50,50); // rueda roja

fill(#D4DB26); // amatillo oscuro

rect(rectX+27+90,122,42,-42); // rect celeste

fill(#FEFF29);// amatillo claro

triangle((width/2)-15,0,(width/2-60),60, (width/2+30), 60); // techo Amarillo

***DISEÑO:***

|  |  |
| --- | --- |
| **ENTIDAD**: LIENZO | |
| **VARIABLES:** | |
| anchoLienzo, altoLienzo anchX, altoY: enteros | |
| Nombre algoritmo: dibujar\_Puntos\_Aleatorios\_de\_colores  **ALGORITMO:**  Inicio  anchoLienzo <- 600;  altoLienzo <-300;  float X;  float Y;  Dibujar  X= width ;  Y= height;  float rectX= X/4; //rectangulo en X  float rectY= Y-2; //rectangulo en Y  rect(rectX, 0 ,rectY-100, Y); // cuadro del fondo  fill(#FF0303); //relleno rojo  rect(rectX+25,rectY-100,30,-100);  fill(#09D636);  rect(rectX+25,rectY-50,153,-100); //VERDE - rectX,rectX - 50);  fill(#2DD8ED);  rect(rectX+25+80,rectY-120,65,-120); // cabina ceñeste  ellipse(rectX+50+2,rectY-40,50,50);// rueda celeste  fill(#FF0303);  ellipse(rectX+150+2,rectY-40,50,50); // rueda roja  fill(#D4DB26); // amatillo oscuro  rect(rectX+27+90,122,42,-42); // rect celeste  fill(#FEFF29);// amatillo claro  triangle((width/2)-15,0,(width/2-60),60, (width/2+30), 60); // techo amarillo  };  FIN. | |
|  | |

***CODIFICACION:***

***Ejercicio 8:*** Trate de replicar el siguiente dibujo:

***DESCRIPCION DEL PORBLEMA***; Dibujar en Processing puntos de colores aleatorios y posicion con random.

***ANALASIS:***

***DATOS DE ENTRADA***

lienzo (400,400).

***DATOS DE SALIDA***

Dibujar un tren de colores con primitivas en el lienzo

***PORCESO***

float X;

float Y;

Dibujar

X= width ;

Y= height;

float rectX= X/4; //rectangulo en X

float rectY= Y-2; //rectangulo en Y

rect(rectX, 0 ,rectY-100, Y); // cuadro del fondo

fill(#FF0303); //relleno rojo

rect(rectX+25,rectY-100,30,-100);

fill(#09D636);

rect(rectX+25,rectY-50,153,-100); //VERDE - rectX,rectX - 50);

fill(#2DD8ED);

rect(rectX+25+80,rectY-120,65,-120); // cabina ceñeste

ellipse(rectX+50+2,rectY-40,50,50);// rueda celeste

fill(#FF0303);

ellipse(rectX+150+2,rectY-40,50,50); // rueda roja

fill(#D4DB26); // amatillo oscuro

rect(rectX+27+90,122,42,-42); // rect celeste

fill(#FEFF29);// amatillo claro

triangle((width/2)-15,0,(width/2-60),60, (width/2+30), 60); // techo Amarillo

***DISEÑO:***

|  |
| --- |
| **ENTIDAD**: LIENZO |
| **VARIABLES:** |
| anchoLienzo, altoLienzo anchX, altoY: enteros |
| Nombre algoritmo: dibujar\_Puntos\_Aleatorios\_de\_colores  **ALGORITMO:**  Inicio  anchoLienzo <- 600;  altoLienzo <-300;  float X;  float Y;  Dibujar  X= width ;  Y= height;  float rectX= X/4; //rectangulo en X  float rectY= Y-2; //rectangulo en Y  rect(rectX, 0 ,rectY-100, Y); // cuadro del fondo  fill(#FF0303); //relleno rojo  rect(rectX+25,rectY-100,30,-100);  fill(#09D636);  rect(rectX+25,rectY-50,153,-100); //VERDE - rectX,rectX - 50);  fill(#2DD8ED);  rect(rectX+25+80,rectY-120,65,-120); // cabina ceñeste  ellipse(rectX+50+2,rectY-40,50,50);// rueda celeste  fill(#FF0303);  ellipse(rectX+150+2,rectY-40,50,50); // rueda roja  fill(#D4DB26); // amatillo oscuro  rect(rectX+27+90,122,42,-42); // rect celeste  fill(#FEFF29);// amatillo claro  triangle((width/2)-15,0,(width/2-60),60, (width/2+30), 60); // techo amarillo  };  FIN. |
|  |

onclusión

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)