Trabajo Practico N01: Algoritmos y Estructuras de Control

Ejercicio 01: Evaluar (obtener resultado) la siguiente expresión para A = 2 y B = 5

3\*A – 4\*B / A^2

Resolución necesaria en Word:

3\*2 – 4\*5 / 2^2

6 – 20 / 4

6 – 5 .

1 .

1. Resolvemos primero las potencias y multiplicaciones
2. Resolvemos la división
3. Restamos
4. Resultado

Fase de análisis:

* Especificación del Problema:

Dado dos números, obtener el resultado de la ecuación.

* Análisis:

Datos de Entrada: A, B - Entero

Datos de Salida: res - Flotante

* Proceso:

// resolver la ecuación con los números dados

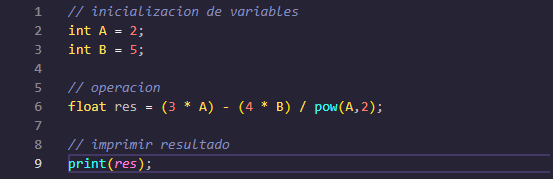
3\*A – 4\*B / A^2 // devuelve un valor flotante

// al resolverla hay que mostrar el resultado por la pantalla

res = 1

Fase de diseño:

|  |
| --- |
| ENTIDAD QUE RESUEVE EL PROBLEMA: Programa |
| VARIABLES   1. A, B: entero 2. res: flotante |
| NOMBRE DEL ALGORITMO: Resolver operación  PROCESO DEL ALGORITMO:   1. Leer A, B 2. Calcular 3\*A – 4\*B / A^2 3. Resultado res 4. Mostrar res |



Ejercicio 02: Evaluar la siguiente expresión:

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

Resolución necesaria en Word:

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

4 / 2 \* 3 / 6 + 6 / 2 / 1 / 25 / 4 \* 2 .

2 \* 0.5 + 3 / 25 / 4 \* 2

1 + 0.12 / 4 \* 2

1 + 0.06 .

1.06

1. Separamos por términos y resolvemos primero las potencias
2. Resolvemos las divisiones y multiplicaciones de izquierda a derecha
3. Resolvemos las multiplicaciones y divisiones restantes
4. Sumamos
5. Resultado

Fase de análisis:

* Especificación del Problema:

Obtener el resultado del problema matemático.

* Análisis:

Datos de Salida: res - Flotante

* Proceso:

// resolver el problema matemático

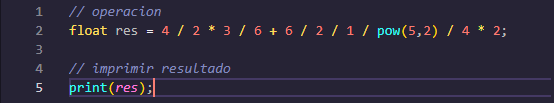
4 / 2 \* 3 / 6 + 6 / 2 / 1 / 5^2 / 4 \* 2 // devuelve un valor flotante

// al resolverlo hay que mostrar el resultado por la pantalla

res = 1.06

Fase de diseño:

|  |
| --- |
| ENTIDAD QUE RESUEVE EL PROBLEMA: Programa |
| VARIABLES   1. res: flotante |
| NOMBRE DEL ALGORITMO: Resolver operación  PROCESO DEL ALGORITMO:   1. Calcular 4 / 2 \* 3 / 6 + 6 / 2 / 1 / 5^2 / 4 \* 2 2. Resultado res 3. Mostrar res |



Ejercicio 05: Si A = 4, B = 5 y C = 1, evaluar las siguientes expresiones:

1. B \* A – B ^ 2 / 4 \* C

Resolución necesaria en Word:

5 \* 4 – 5 ^2 / 4 \* 1

20 – 25 / 4 .

20 – 6.25 .

13.75 .

1. Separamos por términos y resolvemos primero las potencias y multiplicaciones
2. Resolvemos las divisiones
3. Restamos
4. Resultado

Fase de análisis:

* Especificación del Problema:

Dado tres números, obtener el resultado de la ecuación.

* Análisis:

Datos de Entrada: A, B, C - entero

Datos de Salida: res - Flotante

* Proceso:

// resolver la ecuación con los números dados

B \* A – B ^ 2 / 4 \* C // devuelve un valor flotante

// al resolverla hay que mostrar el resultado por la pantalla

res = 13.75

Fase de diseño:

|  |
| --- |
| ENTIDAD QUE RESUEVE EL PROBLEMA: Programa |
| VARIABLES   1. A, B, C: entero 2. res: flotante |
| NOMBRE DEL ALGORITMO: Resolver operación  PROCESO DEL ALGORITMO:   1. Leer A, B, C 2. Calcular B \* A – B ^ 2 / 4 \* C 3. Resultado res 4. Mostrar res |

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

Resolución necesaria en Word:

(4 \* 5) / 3 ^ 2

20 / 9

2.222̑̑ .

1. Resolvemos la multiplicación y la potencia
2. Resolvemos la división
3. Resultado

Fase de análisis:

* Especificación del Problema:

Dado tres números, obtener el resultado de la ecuación.

* Análisis:

Datos de Entrada: A, B - entero

Datos de Salida: res - Flotante

* Proceso:

// resolver la ecuación con los números dados

(A \* B) / 3 ^ 2 // devuelve un valor flotante

// al resolverla hay que mostrar el resultado por la pantalla

res = 2.222̑̑

Fase de diseño:

|  |
| --- |
| ENTIDAD QUE RESUEVE EL PROBLEMA: Programa |
| VARIABLES   1. A, B: entero 2. res: flotante |
| NOMBRE DEL ALGORITMO: Resolver operación  PROCESO DEL ALGORITMO:   1. Leer A, B 2. Calcular (A \* B) / 3 ^ 2 3. Resultado res 4. Mostrar res |

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

Resolución necesaria en Word:

(((5 + 1) / 2 \* 4 + 10) \* 3 \* 5) – 6

(( 6 / 2 \* 4 + 10) \* 3 \* 5) – 6

(( 12 + 10) \* 3 \* 5) – 6

( 22 \* 3 \* 5) – 6

330 – 6 .

324

1. Separamos por términos y resolvemos primero las los paréntesis
2. Resolvemos las divisiones y multiplicaciones de izquierda a derecha
3. Resolvemos la suma
4. Resolvemos las multiplicaciones
5. Restamos
6. Resultado

Fase de análisis:

* Especificación del Problema:

Dado tres números, obtener el resultado de la ecuación.

* Análisis:

Datos de Entrada: A, B, C - entero

Datos de Salida: res - Flotante

* Proceso:

// resolver la ecuación con los números dados

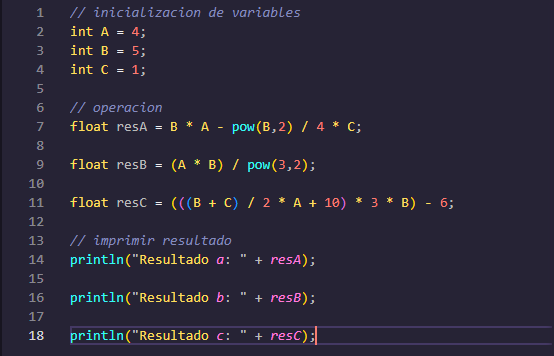
(((B + C) / 2 \* A + 10) \* 3 \* B) – 6// devuelve un valor flotante

// al resolverla hay que mostrar el resultado por la pantalla

res = 324

Fase de diseño:

|  |
| --- |
| ENTIDAD QUE RESUEVE EL PROBLEMA: Programa |
| VARIABLES   1. A, B, C: entero 2. res: flotante |
| NOMBRE DEL ALGORITMO: Resolver operación  PROCESO DEL ALGORITMO:   1. Leer A, B, C 2. Calcular (((B + C) / 2 \* A + 10) \* 3 \* B) – 6 3. Resultado res 4. Mostrar res |



Ejercicio 06: Para X = 3, Y = 4 y Z = 1, evaluar el resultado de:

R1 = Y + Z

R2 = X >= R1

Resolución necesaria en Word:

R1 = 4 + 1

R1 = 5

R2 = 3 >= 5

R2 = false

1. Resolvemos la suma
2. Resultado
3. Comparamos
4. Resultado

Fase de análisis:

* Especificación del Problema:

Dado tres números, obtener el resultado de la ecuación y comparación.

* Análisis:

Datos de Entrada: X, Y, Z - entero

Datos de Salida: res - Flotante

* Proceso:

// resolver la ecuación con los números dados

R1 = Y + Z

// devuelve un valor flotante

// resolver la comparación

R2 = X >= R1

// devuelve un valor booleano

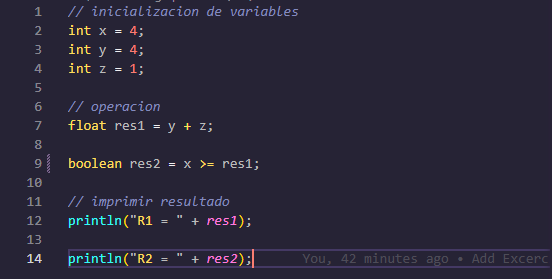
// al resolverla hay que mostrar el resultado por la pantalla

res1 = 5

res2 = false

Fase de diseño:

|  |
| --- |
| ENTIDAD QUE RESUEVE EL PROBLEMA: Programa |
| VARIABLES   1. X, Y, Z: entero 2. res1: flotante 3. res2: booleano |
| NOMBRE DEL ALGORITMO: Comparación  PROCESO DEL ALGORITMO:   1. Leer X, Y, Z 2. Calcular R1 = Y + Z 3. R2 = X >= R1 4. Resultado res1 5. Mostrar res1 , res2 |



Ejercicio 07: Para cont1 = 3, cont2 = 4, evaluar el resultado de:

R1 = ++cont1

R2 = cont1 < cont2

Resolución necesaria en Word:

R1 = ++cont1

R1 = 4

R2 = 4 < 4

R2 = false

1. Resolvemos la suma
2. Resultado
3. Comparamos
4. Resultado

Fase de análisis:

* Especificación del Problema:

Dado dos números, obtener el resultado de la ecuación y comparación.

* Análisis:

Datos de Entrada: cont1, cont2- entero

Datos de Salida: res - Flotante

* Proceso:

// resolver la ecuación con los números dados

R1 = ++cont1

// devuelve un valor flotante

// resolver la comparación

R2 = cont1 < cont2

// devuelve un valor booleano

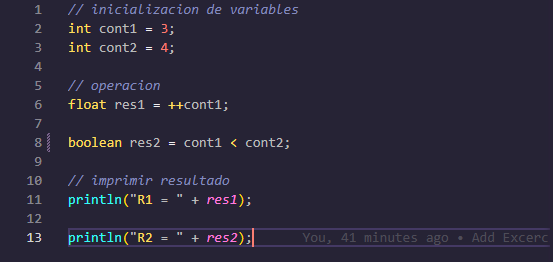
// al resolverla hay que mostrar el resultado por la pantalla

res1 = 5

res2 = false

Fase de diseño:

|  |
| --- |
| ENTIDAD QUE RESUEVE EL PROBLEMA: Programa |
| VARIABLES   1. cont1, cont2: entero 2. res1: flotante 3. res2: booleano |
| NOMBRE DEL ALGORITMO: Comparación  PROCESO DEL ALGORITMO:   1. Leer con1, cont2 2. Calcular R1 = ++cont1 3. Calcular R2 = cont1 < cont2 4. Resultado res1 5. Mostrar res1 , res2 |



Ejercicio 08: Para A = 31, B = – 1, X = 3 e Y = 2, evaluar el resultado de:

A + B – 1 < X \* Y

Resolución necesaria en Word:

31 + (– 1) – 1 < 3 \* 2

29 < 6

false

1. Resolvemos la suma y multiplicación
2. Comparamos
3. Resultado

Fase de análisis:

* Especificación del Problema:

Dado tres números, obtener el resultado de la ecuación y comparación.

* Análisis:

Datos de Entrada: A, B, X, Y - entero

Datos de Salida: res - booleano

* Proceso:

// resolver la ecuación con los números dados

res = A + B – 1 < X \* Y // devuelve un valor flotante

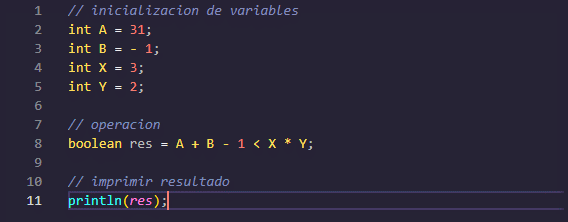
// devuelve un valor booleano

// al resolverla hay que mostrar el resultado por la pantalla

res = false

Fase de diseño:

|  |
| --- |
| ENTIDAD QUE RESUEVE EL PROBLEMA: Programa |
| VARIABLES   1. A, B, X, Y entero 2. res: booleano |
| NOMBRE DEL ALGORITMO: Resolver operación  PROCESO DEL ALGORITMO:   1. Leer A, B, X, Y 2. Calcular res 3. Resultado res 4. Mostrar res |



Ejercicio 09: Para X = 6 e Y = 8, evaluar el resultado de:

!(X < 5) && !(Y >= 7)

Resolución necesaria en Word:

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

!(false) && !(true)

False

1. Comparamos
2. Resolvemos la operación AND
3. Resultado

Fase de análisis:

* Especificación del Problema:

Dados dos números obtener la comparación lógica.

* Análisis:

Datos de Entrada: X, Y - entero

Datos de Salida: res - booleano

* Proceso:

// resolver la ecuación y comparar con los números dados

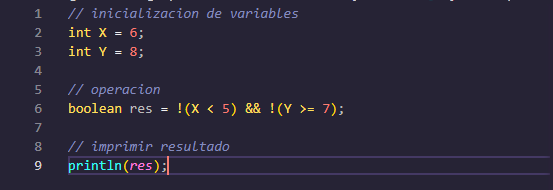
res = !(X < 5) && !(Y >= 7) // devuelve un valor booleano

// al resolverla hay que mostrar el resultado por la pantalla

res = false

Fase de diseño:

|  |
| --- |
| ENTIDAD QUE RESUEVE EL PROBLEMA: Programa |
| VARIABLES   1. X, Y entero 2. res: booleano |
| NOMBRE DEL ALGORITMO: Resolver operación  PROCESO DEL ALGORITMO:   1. Leer X, Y 2. Calcular res 3. Resultado res 4. Mostrar res |



Ejercicio 10: Para I = 22 y J = 3, evaluar el resultado de:

!((I > 4) || !(J <= 6))

Resolución necesaria en Word:

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

!((true) || !(true))

false

1. Comparamos
2. Resolvemos la operación OR
3. Resultado

Fase de análisis:

* Especificación del Problema:

Dados dos números obtener la comparación lógica.

* Análisis:

Datos de Entrada: I, J - entero

Datos de Salida: res - booleano

* Proceso:

// resolver la ecuación y comparar con los números dados

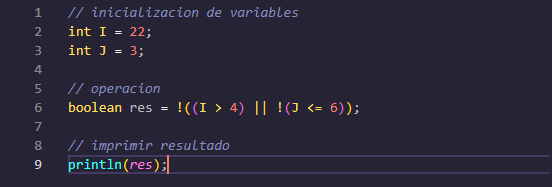
res = !(X < 5) && !(Y >= 7) // devuelve un valor booleano

// al resolverla hay que mostrar el resultado por la pantalla

res = false

Fase de diseño:

|  |
| --- |
| ENTIDAD QUE RESUEVE EL PROBLEMA: Programa |
| VARIABLES   1. I, J entero 2. res: booleano |
| NOMBRE DEL ALGORITMO: Resolver operación  PROCESO DEL ALGORITMO:   1. Leer I, J 2. Calcular !((I > 4) || !(J <= 6)) 3. Resultado res 4. Mostrar res |



Ejercicio 11: Para A = 34, B = 12 y C = 8, evaluar el resultado de:

!( A + B == C) || (C != 0) && (B – C >= 19)

Resolución necesaria en Word:

!( 34+12==8) || (8!=0) && (12–8>=19)

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

!(false) || true && false

true && false

false

1. Resolvemos las sumas y restas
2. Comparamos
3. Resolvemos la operación OR
4. Resolvemos la operación AND
5. Resultado

Fase de análisis:

* Especificación del Problema:

Dados dos números obtener la comparación lógica.

* Análisis:

Datos de Entrada: A, B, C - entero

Datos de Salida: res - booleano

* Proceso:

// resolver la ecuación y comparar con los números dados

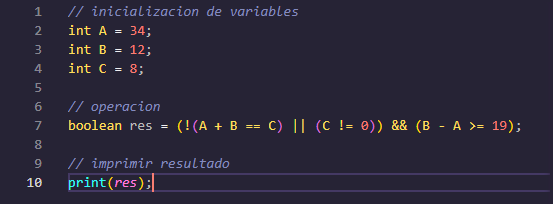
res = !(X < 5) && !(Y >= 7) // devuelve un valor booleano

// al resolverla hay que mostrar el resultado por la pantalla

res = false

Fase de diseño:

|  |
| --- |
| ENTIDAD QUE RESUEVE EL PROBLEMA: Programa |
| VARIABLES   1. A, B, C entero 2. res: booleano |
| NOMBRE DEL ALGORITMO: Resolver operación  PROCESO DEL ALGORITMO:   1. Leer A, B, C 2. Calcular !((I > 4) || !(J <= 6)) 3. Resultado res 4. Mostrar res |



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.

Resolución necesaria en Word:

1. Pedimos al usuario un nombre
2. Creamos el mensaje
3. Mostramos el mensaje con el nombre

Fase de análisis:

* Especificación del Problema:

Imprimir un mensaje con el nombre dado por el usuario.

* Análisis:

Datos de Entrada: name - string

Datos de Salida: saludo - string

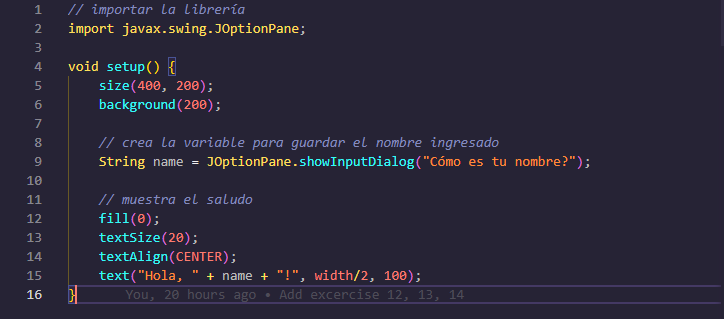
* Proceso:

// pedirle al usuario un nombre

// Mostrar el mensaje por la pantalla

Fase de diseño:

|  |
| --- |
| ENTIDAD QUE RESUEVE EL PROBLEMA: Programa |
| VARIABLES   1. name string 2. saludo string |
| NOMBRE DEL ALGORITMO: Saludo con nombre  PROCESO DEL ALGORITMO:   1. Pedir name 2. Crear mensaje con name 3. Mostrar mensaje |



Ejercicio 13: Será común resolver problemas utilizando variables. Calcule el perímetro y

área de un rectángulo dada su base y su altura.

Resolución necesaria en Word:

1. Pedimos al base y altura
2. Calculamos el perímetro y área
3. Mostramos los resultados

Fase de análisis:

* Especificación del Problema:

Calcular el perímetro y el área de un rectángulo.

* Análisis:

Datos de Entrada: base, altura - float

Datos de Salida: perímetro, área - float

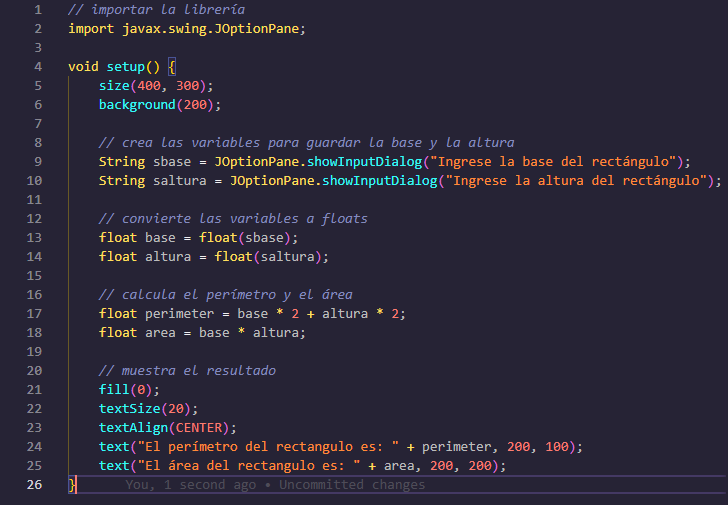
* Proceso:

// pedirle al usuario base y altura

// Mostrar el perímetro y el área por la pantalla

Fase de diseño:

|  |
| --- |
| ENTIDAD QUE RESUEVE EL PROBLEMA: Programa |
| VARIABLES   1. base, altura float 2. perímetro, área float |
| NOMBRE DEL ALGORITMO: Calculo de perímetro y área de un rectangulo  PROCESO DEL ALGORITMO:   1. Pedir base, altura 2. Calcular perímetro y área 3. Mostrar resultado |



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:

Resolución necesaria en Word:

1. Pedimos el cateto opuesto y el adyacente
2. Calculamos la hipotenusa
3. Mostramos el resultado

Fase de análisis:

* Especificación del Problema:

Calcular la hipotenusa.

* Análisis:

Datos de Entrada: cateto opuesto, cateto adyacente - float

Datos de Salida: hipotenusa - float

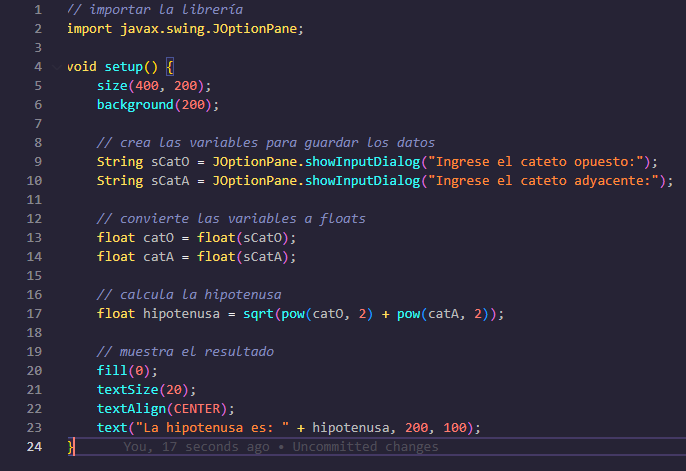
* Proceso:

// pedirle al usuario base y altura

// Mostrar el perímetro y el área por la pantalla

Fase de diseño:

|  |
| --- |
| ENTIDAD QUE RESUEVE EL PROBLEMA: Programa |
| VARIABLES   1. cateto opuesto, cateto adyacente float 2. hipotenusa float |
| NOMBRE DEL ALGORITMO: Calculo de hipotenusa  PROCESO DEL ALGORITMO:   1. Pedir cateto opuesto y adyacente 2. Calcular hipotenusa 3. Mostrar resultado |



Ejercicio 15: 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:

Resolución necesaria en Word:

1. Pedimos numA y numB
2. Realizamos los cálculos
3. Mostramos los resultados

Fase de análisis:

* Especificación del Problema:

Calcular la suma, resta, multiplicación y división de dos números.

* Análisis:

Datos de Entrada: numA, numB - float

Datos de Salida: suma, resta, multiplicación, división - float

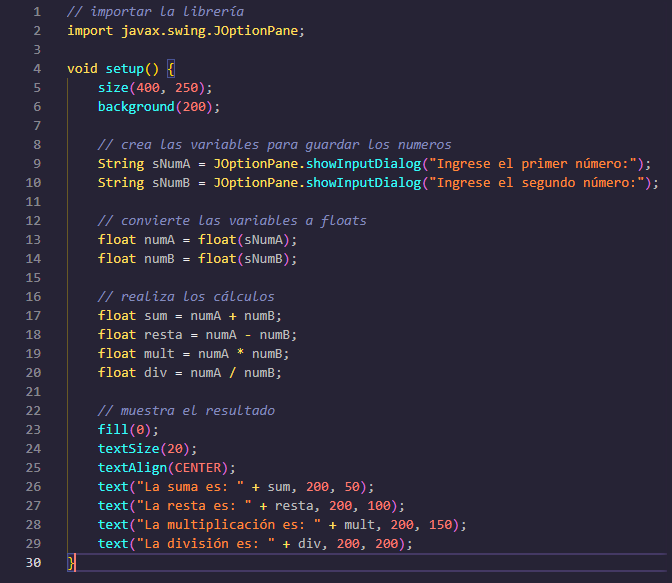
* Proceso:

// pedirle al usuario dos números

// Mostrar el resultado de los cálculos por la pantalla

Fase de diseño:

|  |
| --- |
| ENTIDAD QUE RESUEVE EL PROBLEMA: Programa |
| VARIABLES   1. numA, numB float 2. suma, resta, mult, div float |
| NOMBRE DEL ALGORITMO: Cálculos algebraicos con dos números  PROCESO DEL ALGORITMO:   1. Pedir numA y numB 2. Calcular suma, resta, mult y div 3. Mostrar resultado |



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:

Resolución necesaria en Word:

1. Pedimos la temp en Fahrenheit
2. Realizamos la conversión
3. Mostramos el resultado

Fase de análisis:

* Especificación del Problema:

Convertir temperatura Fahrenheit en Celsius.

* Análisis:

Datos de Entrada: tempF - float

Datos de Salida: tempC - float

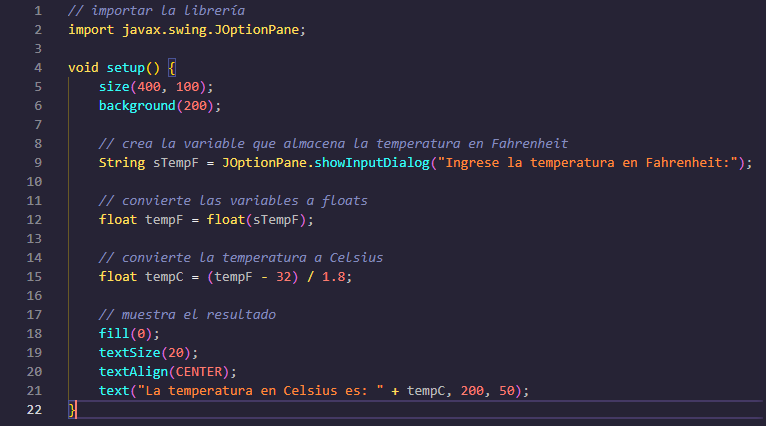
* Proceso:

// pedirle al usuario la temp en Fahrenheit

// Mostrar la temp en Celsius por la pantalla

Fase de diseño:

|  |
| --- |
| ENTIDAD QUE RESUEVE EL PROBLEMA: Programa |
| VARIABLES   1. tempF float 2. tempC float |
| NOMBRE DEL ALGORITMO: Conversor de °F a °C  PROCESO DEL ALGORITMO:   1. Pedir tempF 2. Calcular tempC 3. Mostrar resultado |



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 Link con un Círculo, y al tesoro con un cuadrado. Además, mueva a Link mediante el mouse.

Resolución necesaria en Word:

1. Tomamos las coordenadas de los puntos 1 y 2
2. Realizamos el cálculo de la hipotenusa
3. Mostramos el resultado y los puntos

Fase de análisis:

* Especificación del Problema:

Calcular la distancia entre dos puntos.

* Análisis:

Datos de Salida: dist - float

* Proceso:

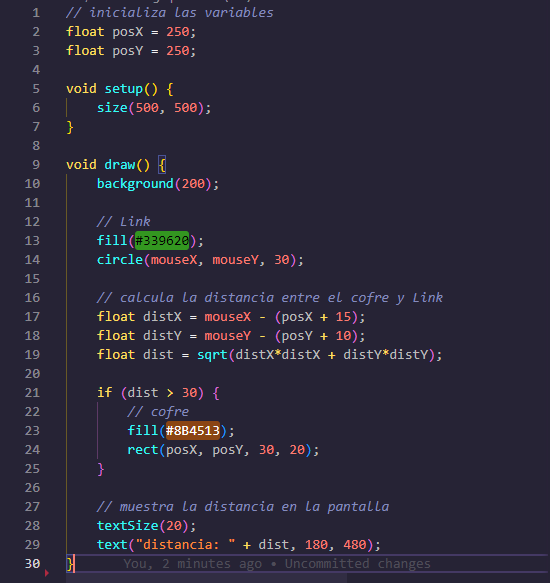
// tomar las coordenadas de dos puntos

// calcular la distancia entre ellos

// mostrar el resultado en pantalla

Fase de diseño:

|  |
| --- |
| ENTIDAD QUE RESUEVE EL PROBLEMA: Programa |
| VARIABLES   1. distX, distY float 2. dist float |
| NOMBRE DEL ALGORITMO: Ocultar al acercar  PROCESO DEL ALGORITMO:   1. Tomar coordenadas del punto 1 y 2 2. Calcular la distancia 3. Mostrar resultado |



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.

Resolución necesaria en Word:

1. Tomamos función cuadrática
2. Realizamos el cálculo de función cuadrática
3. Mostramos las raíces

Fase de análisis:

* Especificación del Problema:

Calcular las raíces de una función.

* Análisis:

Datos de Entrada: funcA, funcB, funcC - string

Datos de Salida: x1, x2 - float

* Proceso:

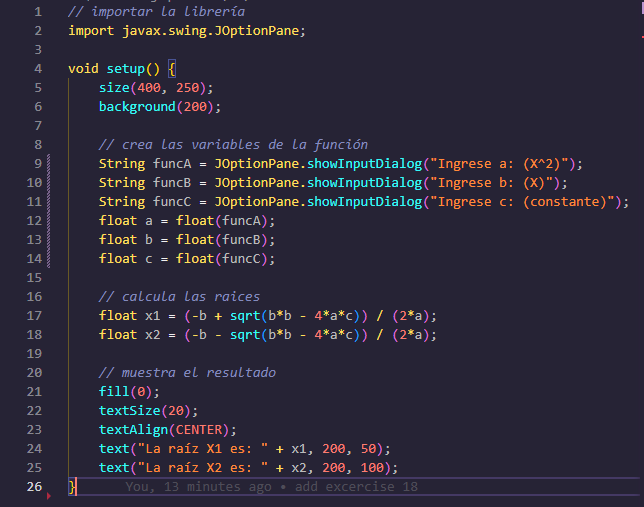
// tomar la función

// calcular las raíces

// mostrar el resultado en pantalla

Fase de diseño:

|  |
| --- |
| ENTIDAD QUE RESUEVE EL PROBLEMA: Programa |
| VARIABLES   1. funcA, funcB, funcC string 2. x1, x2 float |
| NOMBRE DEL ALGORITMO: Raíces de una función  PROCESO DEL ALGORITMO:   1. Tomar los términos 2. Calcular las raíces 3. Mostrar el resultado |



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.

Resolución necesaria en Word:

1. Definimos la altura
2. Dibujamos la línea y el círculo
3. Movemos de arriba hacia abajo y viceversa

Fase de análisis:

* Especificación del Problema:

Mover dos figuras de arriba hacia abajo y viceversa.

* Análisis:

Datos de Salida: movimiento ascendente y descendente - float

* Proceso:

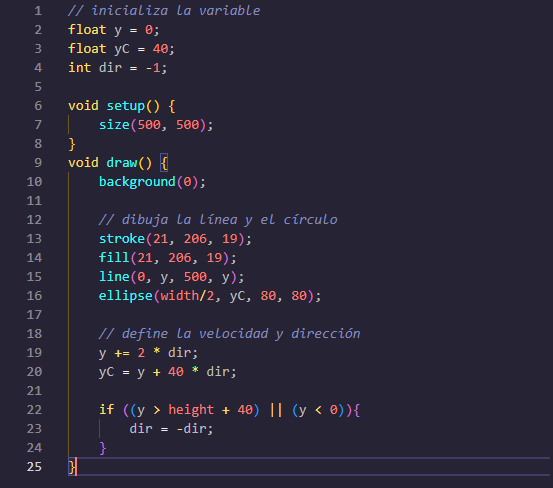
// definir altura

// dibujar la línea y el circulo

// mover ascendente y descendentemente

Fase de diseño:

|  |
| --- |
| ENTIDAD QUE RESUEVE EL PROBLEMA: Programa |
| VARIABLES   1. y, yC float |
| NOMBRE DEL ALGORITMO: Pelota con movimiento vertical  PROCESO DEL ALGORITMO:   1. Definir variables 2. Dibujar línea y circulo 3. Mover para arriba y abajo |



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.

Resolución necesaria en Word:

1. Tomamos el alto y ancho de un rectángulo
2. Acomodamos los rectángulos en la pantalla

Fase de análisis:

* Especificación del Problema:

Mostrar los rectángulos de forma ordenada.

* Análisis:

Datos de Salida: ubicaciones del rectángulo

* Proceso:

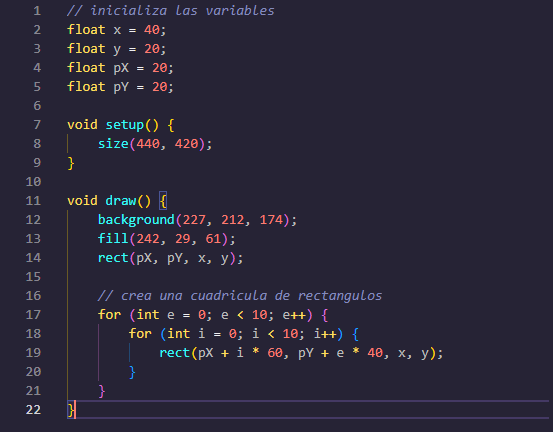
// tomar un rectángulo

// posicionarlo en la pantalla

// completar la pantalla con rectángulos

Fase de diseño:

|  |
| --- |
| ENTIDAD QUE RESUEVE EL PROBLEMA: Programa |
| VARIABLES   1. x, y float |
| NOMBRE DEL ALGORITMO: Rectángulos ordenados  PROCESO DEL ALGORITMO:   1. tomar las medidas de un rectángulo 2. Acomodarlo a o largo de la pantalla |



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.

Resolución necesaria en Word:

1. Definimos el las medidas del escalón
2. Acomodamos los escalones en la pantalla

Fase de análisis:

* Especificación del Problema:

Mostrar una escalera en pantalla.

* Análisis:

Datos de Salida: escalera ordenada.

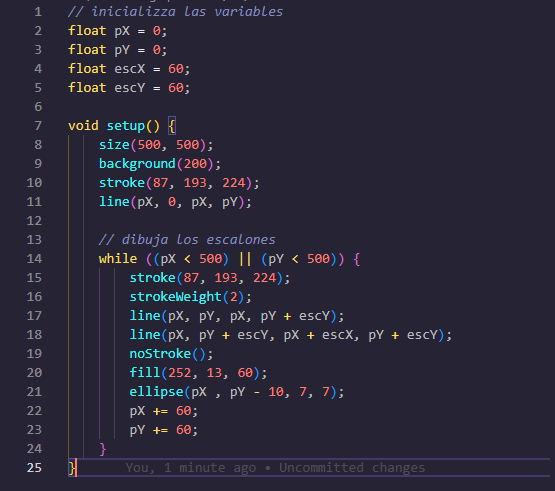
* Proceso:

// Inicializa las variables

// muestra los escolanes en la pantalla

Fase de diseño:

|  |
| --- |
| ENTIDAD QUE RESUEVE EL PROBLEMA: Programa |
| VARIABLES   1. pX, pY, escX, escY float |
| NOMBRE DEL ALGORITMO: Escalón con puntos  PROCESO DEL ALGORITMO:   1. Define el escalón 2. Acomodar escalera en la pantalla |



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.

Resolución necesaria en Word:

1. Definimos las líneas horizontales.
2. Pintamos los círculos de distintos colores mientras se dibujan.
3. Acomodamos los círculos sobre las líneas.

Fase de análisis:

* Especificación del Problema:

Mostrar líneas y círculos de distintos colores con líneas de por medio.

* Análisis:

Datos de Salida: líneas y círculos de distintos colores.

* Proceso:

// Inicializa las variables

// dibuja las líneas

// dibuja los círculos

Fase de diseño:

|  |
| --- |
| ENTIDAD QUE RESUEVE EL PROBLEMA: Programa |
| VARIABLES   1. pX, pY float |
| NOMBRE DEL ALGORITMO: Líneas y círculos de por medio.  PROCESO DEL ALGORITMO:   1. Define las líneas 2. Dibuja los círculos pintados |

