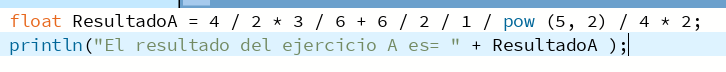
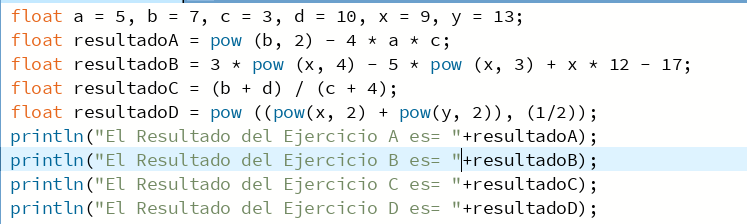
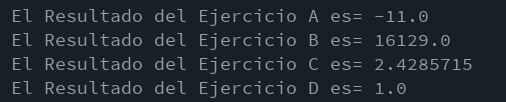
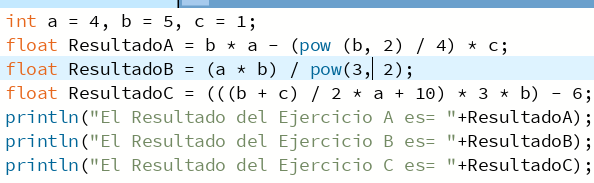
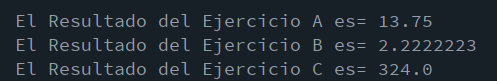
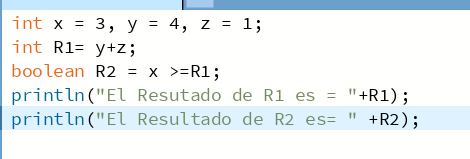
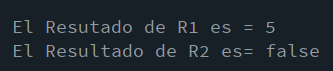
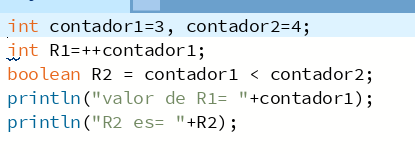
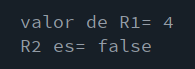
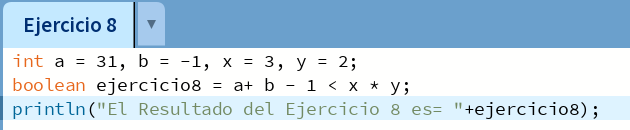
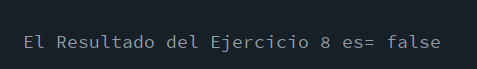
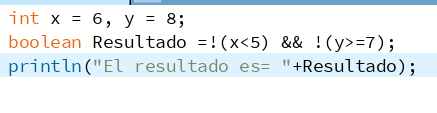
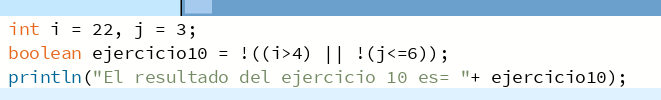
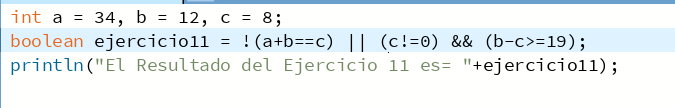
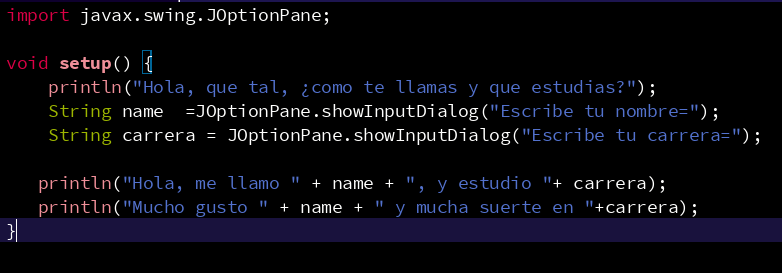
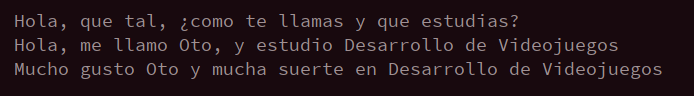
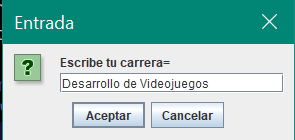
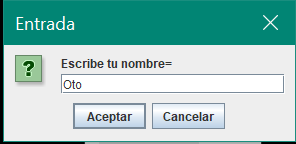
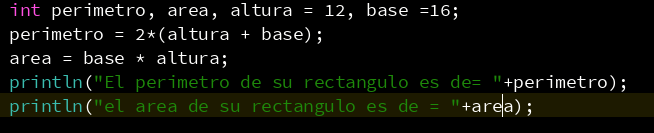
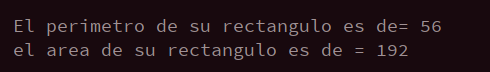
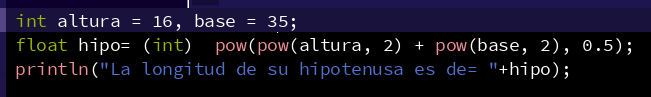
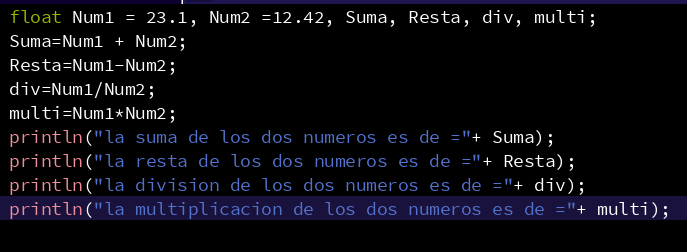
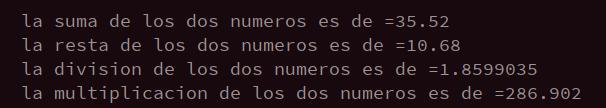
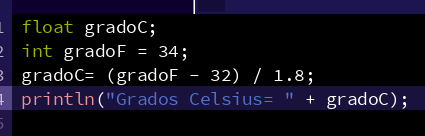
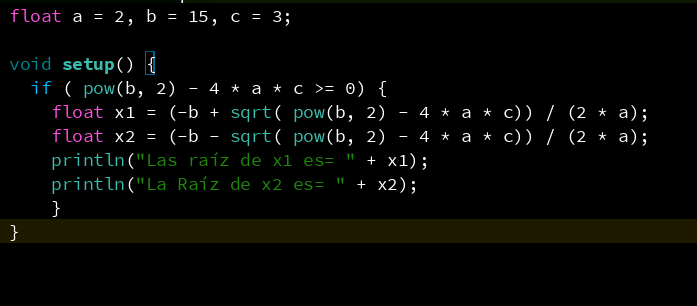
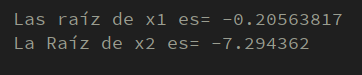
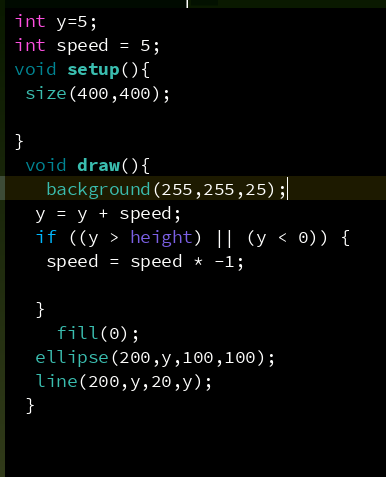
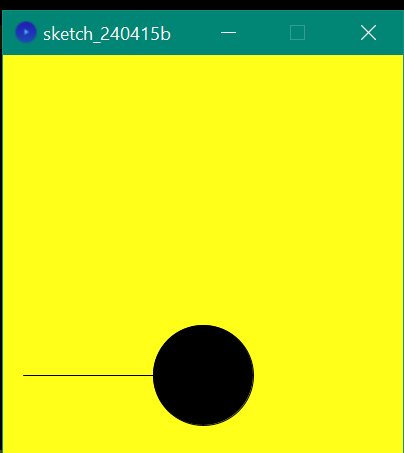
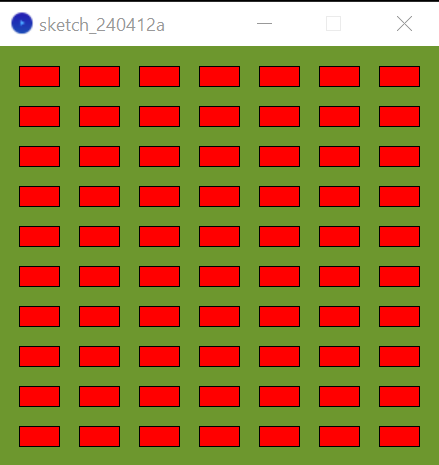
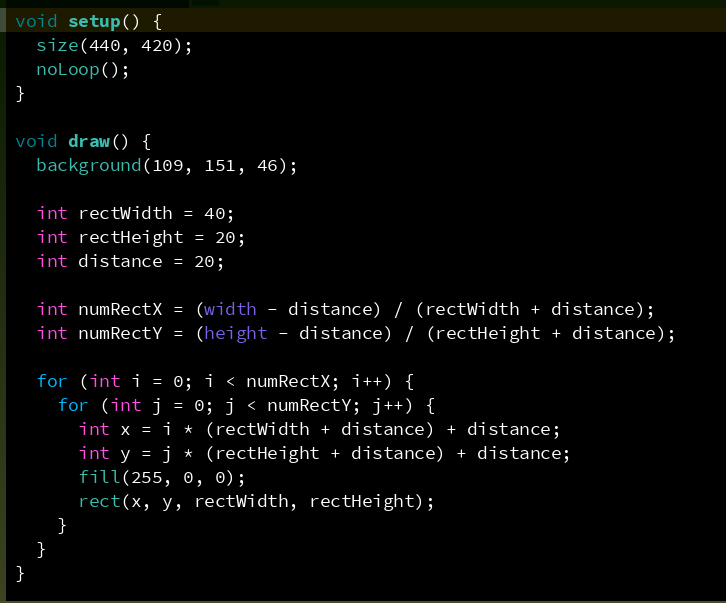
**Fundamentos de Programación Orientada a Objetos  
Tema: Trabajo Práctico N°1: Operadores – Metodología de Programación  
Nombre: Mateo Otoniel Farfan Boxler DNI: 44229186**  
  
**Ejercicio 2: Evaluar la siguiente expresión 4 / 2 \* 3 / 6 + 6 / 2 / 1 / 5 ^ 2 / 4 \* 2**  
4 / 2 \* 3 / 6 + 6 / 2 / 1 / 5 ^ 2 / 4 \* 2  
4 / 2 \* 3 / 6 + 6 / 2 / 1 / 25 / 4 \* 2  
2 \* 3 / 6 + 3 / 1 / 25 / 4 \* 2  
1 + 0,12 / 4 \* 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.**   
No esta…  
  
**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.**  
a) b ^ 2 – 4 \* a \* c  
7 − 4. 5. 3  
  
b) 3 \* X ^ 4 – 5 \* X ^ 3 + X 12 – 17  
3. 9 – 5. 9 + 9.12 – 17  
  
c) (b + d) / (c + 4)  
7+10  
3+4  
  
d) (x ^ 2 + y ^ 2) ^ (1 / 2)  
  
9 + 13   
  
  
  
  
  
  
**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:  
a) B \* A – B ^ 2 / 4 \* C**5 \* 4 – 5 ^ 2 / 4 \* 1  
20 – 25 / 4   
20 – 6,25  
13,75  
  
b) (A \* B) / 3 ^ 2   
(4 \* 5) / 3 ^ 2  
20 / 9  
2,22  
  
c) (((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  
((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   
4 + 1 = 5  
R2 = x >= R1  
3 >= 5 = Falso  
  
  
  
**Ejercicio 7: Para contador1=3, contador3=4, evaluar el resultado de**   
R1 = ++contador1   
R2 = contador1 < contador2  
  
  
  
  
**Ejercicio 8: Para a=31, b=-1; x=3, y=2, evaluar el resultado de**   
a+b-1 < x\*y  
  
31+(-1)-1 < 3 \* 2  
 29 < 6  
R: False  
  
  
  
**Ejercicio 9: Para x=6, y=8, evaluar el resultado de**   
!(x<5)CC !(y>=7)  
  
  
  
**Ejercicio 10: Para i=22, j=3, evaluar el resultado de !((i>4) || !(j<=6))**  
  
  
  
  
**Ejercicio 11: Para a=34, b=12,c=8, evaluar el resultado de**   
!(a+b==c) || (c!=0)CC(b-c>=19)  
  
  
  
**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.**  
**Analisis**  
**Datos de entrada:**   
name (cadena)  
carrera(cadena)  
**datos de salida:**  
 mensaje – saludo(cadena de texto)  
mensaje-carrera(cadena de texto)  
**Quien realiza el proceso:** Computadora.  
**Cual es el proceso**: Ingresar nombre y carrera que devuelve un saludo con el nombre y la carrera que puso el usuario.  
  
**Diseño**  
**Entidad que resuelve: computadora.**  
**Variables:**  
name(String) – almacena nombre  
carrera(String)- almacena carrera  
  
**Proceso:**  
Inicio  
Leer nombre y carrera  
saludo y carrera <- mucho gusto “ name” mucha suerte en “Carrera”  
mostrar saludo  
Fin  
  
  
  
  
  
**Ejercicio 13: Será común resolver problemas utilizando variables. Calcule el perímetro y área de un rectángulo dada su base y su altura.**  
**Datos de entrada:**  
base, altura (float)  
**datos de salida:** perímetro, área  
proceso  
**Quien realiza el proceso**: calculadora  
**que proceso lo resuelve**:  
Calcula el perímetro y área de un rect. Utilizando la formula adecuada  
P = 2 (base + altura)   
 A = base \* altura  
**Diseño**  
**Entidad que resuelve: computadora**  
**Variable**  
perímetro = int  
área = int  
altura = int  
base= int  
**Proceso del Algoritmo**  
Inicio  
Leer perímetro, área, altura, base  
perímetro = 2 \* (altura + base)  
área = base \* altura  
El perímetro es : + perímetro-  
el área es: + área  
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  
Analisis**  
  
**Datos de entrada**  
Altura, Base  
**Datos de salida**  
Hipotenusa  
**Quien lo realiza:** Computadora  
**Cual es el proceso:** Para obtener la hipotenusa del triangulo rectángulo, se debe obtener los datos de los catetos, una vez obtenemos estos, se aplica la formula: h^2 = a ^ 2 + b ^ 2  
**Diseño**  
altura y base = int  
hipotenusa = float  
  
**Proceso**  
inicio  
leer altura  
leer base  
hipo = *(a^2 + b^2 ) ^(0.5)*  
Mostrar hipotenusa   
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.**

A=5  
B=7  
C=3  
D=10  
X=9  
Y=13

A=4  
B=5  
C=1

**Analisis  
Datos de Entrada**  
Num1, Num2  
**Datos de salida**  
Resultados de suma, resta, div, multiplicación de num1 y num2.  
**Quien realiza el proceso**: computadora  
**Cual es el proceso**:  
  
**Variables:**   
num1, num2 = float  
suma, resta, div, multiplicación = float  
Proceso  
Inicio  
Leer num1, num2  
Suma = num1 + num2  
resta = num1 – num2   
div = num1 / num2  
multiplicación = num1 / num2  
La de suma es… + suma  
La resta es… + resta  
la división es… + div  
la multiplicación es… + multi  
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**  
  
**Datos de entrada**:  
Temperatura en grados Fahrenheit  
**Datos de salida**  
Temperatura en grados Celsius  
  
**Quien realiza el proceso**: Calculadora  
  
**Cual es el proceso**: El proceso consiste en que hay que hacer una conversión de grados Fahrenheit a grados Celsius a través de una formula obtenida. Celsius = (Fahrenheit – 32) / 1.8   
**variables**  
  
gradoC float  
gradoF int  
  
**Proceso**  
  
Inicio  
Leer gradoF  
gradoC = (gradoF - 32) / 1.8  
Mostrar: Grados Celsius … + GradoC  
  
  
  
  
**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.**  
**datos de entrada** :  
a, b y c.  
**Datos de salida**:  
Raices  
**Proceso**  
**Quien realiza el proceso?**  
Computadora  
**Cual es el proceso?**  
Calcular mediante la formula obtenida y asi poder sacar las raíces.  
**Diseño**  
**Entidad que resuelve:** computadora  
a, b y c : float.  
  
**Proceso**:  
Inicio  
Leer a, b y c  
si b^2 – 4 \* a \* c >= 0  
raíz1 = –b + sqrt (pow (b,2) – 4 \* a \* c)) / (2 \* a)   
raíz2 = –b - sqrt (pow (b,2) – 4 \* a \* c)) / (2 \* a)  
mostrar la raíz de x1 es + x1  
mostrar la raíz de x2 es + x2  
finsi  
Fin  
  
  
  
**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**  
**Analisis**   
**Datos de entrada:**   
y , speed  
**datos de salida:**  
Pelota subiendo y bajando con una linea  
**Proceso**   
**Quien es el que realiza el proceso**: Computadora  
**Cual es el Proceso para hallar la solución**:  
  
**Diseño**  
**Entidad que Resuelve el problema** : Lienzo  
**Variables:**  
y, speed = int  
**Proceso del algoritmo:**  
inicio  
leer y, speed  
ancho y alto lienzo = 400, 400  
y = y + speed  
Si ((y > altura ) o (y < 0))  
speed = speed \* -1   
finSi  
color circulo y linea = negro  
dibujar circulo (200,y,100,100)  
dibujar linea(200,y,20,y)  
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:**  
  
  
  
**Analisis  
Datos de entrada**  
Dimensiones del lienzo: ancho = 440, alto = 420.

Medidas del rectángulo: ancho = 40, alto = 20.

Distancia entre rectángulos: 20 píxeles.  
**Datos de Salida**Dibujo del lienzo con los rectángulos solicitados.  
Entidad que resuelve el problema : Lienzo  
**Variables**  
coordenadas Rect: float

ancho, alto, distanciaEntreRect: int

ancholienzo, altoLienzo: int   
inicio

anchoLienzo 🡨440

altoLienzo🡨420

ancho🡨40

alto🡨20

distanicaRect🡨20

para x cooredenadasRect.x hasta anchoLienzo con paso (acho+distanciaEntreRect)

hacer

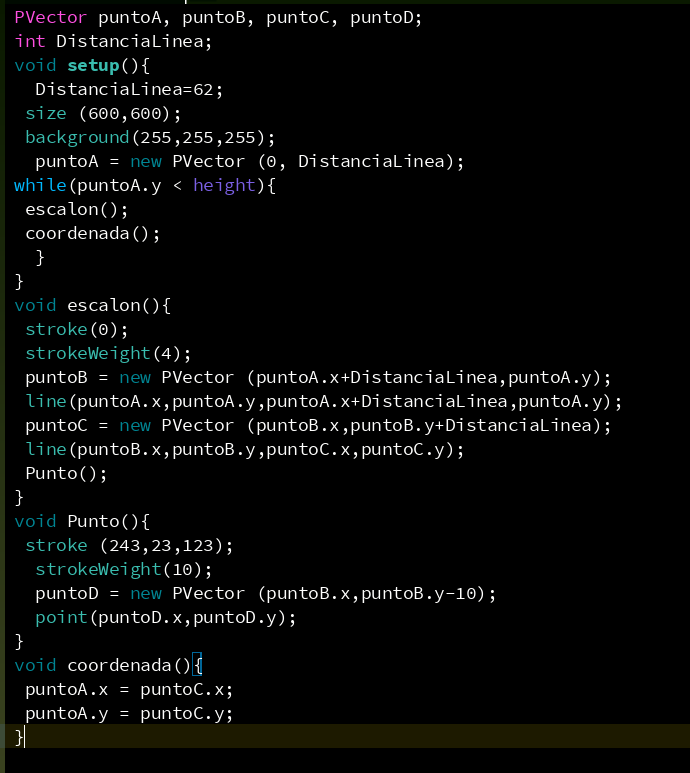
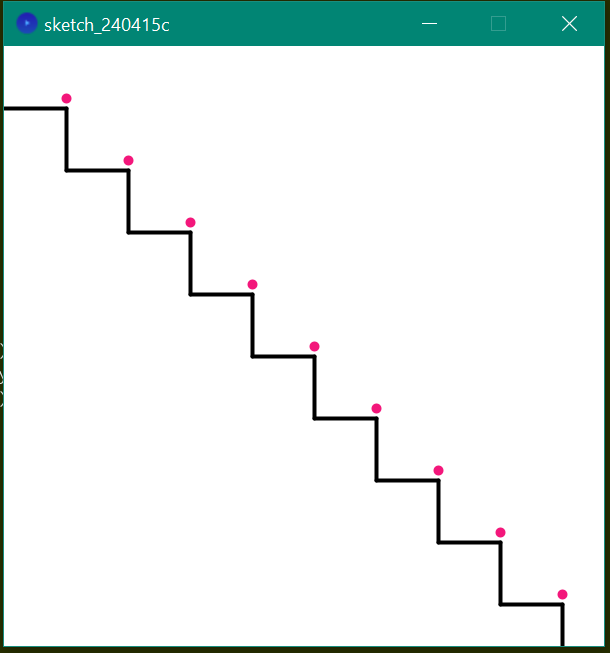
para y =coordenadosRect.y hasta altoLienzo con paso (alto+distanciaEntreRect)

hacer

dibujar rectángulo en (x,y,ancho,alto)

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 de color rojo**  
**Datos de entrada**  
puntoA, puntoB, puntoC, puntoD, DistanciaLinea.  
**Datos de Salida**  
  
Imagen que consiste en Escalones con Puntos sobre si.  
**Quien debe realizar el proceso:** Processing  
  
**Cual es el proceso que resuelve**: Consiste mediante la estructura while, para poder llegar a la solución y dibujar puntos sobre los escalones  
  
**Variables:**   
  
puntoA, puntoB, puntoC, puntoD, int  
DistanciaLinea int  
  
**Proceso del Algoritmo**  
  
inicio  
ancho y alto lienzo = 500,500  
DistanciaLinea = 62  
mientras (puntoA.y <= ancho lienzo ) HACER  
dibujar linea horizontal en (puntoA.x, puntoA.y, puntoB.x, puntoB.y)  
dibujar linea vertixal en (puntoB.x, puntoB.y, puntoC.x, puntoC.y)  
dibujar punto en(puntoD.x, puntoD.y)  
puntoA.x = puntoC.x  
puntoA.y = puntoC.y  
Fin Mientras  
Fin  
  
  
  
**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.  
  
**Analisis**   
  
**Datos de entrada:**  
Numeros de líneas y círculos  
**Datos de Salida:**  
círculos con colores randoms sobre líneas con distanciamiento  
  
**Proceso:**  
**Quien debe realizar el proceso?**  
El lienzo se divide verticalmente de medidas iguales, donde ahí se dibujan líneas en ellas. En cada linea de forma alternativa, se dibujan círculos con colores aleatorios que están espaciados de una misma medida a lo largo de la linea.  
**Cual es el proceso que resuelve?**  
Processing  
**variables:**linX, linY, cirY, distCir = int  
  
**Proceso del Algoritmo**  
Inicio  
ancho y alto de lienzo = 600, 600  
linx = 0  
liny= 100  
cirY = 75  
distCir = 30  
Hacer  
cirX = distCir  
HACER  
dibujar linea en (linX, linY, anchoLienzo, linY)  
dibujar circulo en cirX, cirY, 50, 50  
cirX = cirX + distCir  
Fin Hacer  
MIENTRAS  
cirX sea menor que ancholienzo  
linY = linY + 100  
cirY = CirY + 200  
Fin Hacer  
Mientras (linY sea menor que altoLienzo)  
Fin  
  
