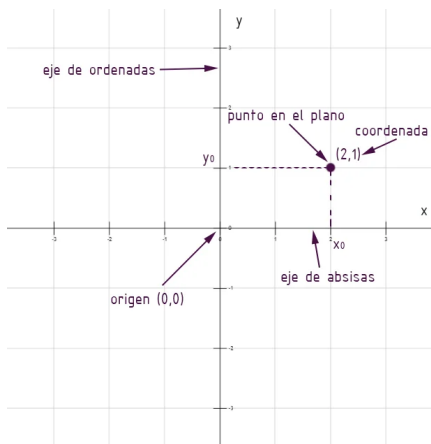


REPRESENTAR UN PLANO EN MATRIZ

REPRESENTAR UN PLANO DE COORDENADAS EN MATRIZ

Muchos problemas están a menudo expresados en un plano de coordenadas clásico de matemáticas.



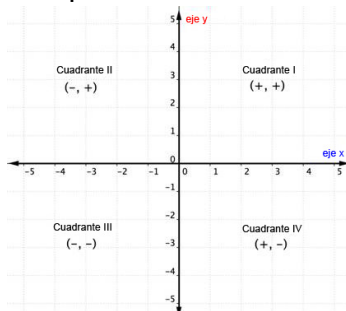
Es posible que queramos almacenar dicho plano en una matriz. Esto es fácil ya que ambos tienen una estructura similar basada en filas y columnas. La idea más importante es que cada par de índices $[i][j]$ se corresponde con una coordenada (x,y) . Luego cada celda de la matriz puede almacenar lo que nos interese: true/false, un String, un int, un objeto, etc...

Piensa en el ejercicio de *Sombras en el camping*. Indican las coordenadas de cada árbol en formato *(columna, fila)*, aunque en este ejercicio era muy fácil "saltar ese escollo", pero hay situaciones más enrevesadas

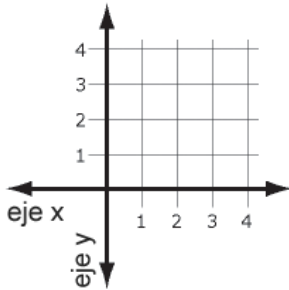
Observa que el eje X matemático se corresponde con las columnas de una matriz y el eje Y con las filas, por ejemplo en notación matemática el punto $(3,2)$ se refiere al cruce de la columna 3 con la fila 2. En cambio, en programación con la sintaxis de índices, expresamos primero las filas y luego las columnas, de forma que la celda de una matriz $[3][2]$ se refiere a la que se almacena en la fila 3 y columna 2 ¡al revés que en el plano de coordenadas!.

Por lo tanto, si el enunciado se expresa con el formato clásico matemático de *(columna, fila)*, estaré atento cuando corresponda a escribirlo $[fila][columna]$ al acceder al array.

La correspondencia entre coordenadas de un plano y de una matriz puede ser más compleja ya que un plano de coordenadas se puede usar con hasta 4 cuadrantes, aunque a veces se usan sólo 1, 2 o 3



Almacenar un plano de coordenadas de un cuadrante en una matriz



Necesitaremos una matriz con tantas celdas como puntos tenga el plano. Por ejemplo, si quisiera representar el plano del dibujo de arriba tengo 5 posibles valores de columnas en eje x (de 0 a 4) y similar en eje Y. Por lo tanto el plano puede representar 5x5 puntos así que necesitaré una matriz con el mismo número de celdas

Ejercicio U4_B4D_E1: Escribir la clase *PlanoJardin* que representa el plano de un jardín para representar donde se van a plantar árboles. El jardín se considera dividido en hileras verticales y horizontales y los árboles se pueden plantar en el cruce de ambas. Almacenamos la plantación de árboles en una matriz pero ojo porque el usuario de la clase desea utilizar el sistema clásico de las coordenadas matemáticas (x,y) para indicar la ubicación de un árbol. La clase es capaz de imprimir un pequeño plano tal cual lo dibuja el usuario en su plano de coordenadas x,y.

Debe funcionar el siguiente main y producir la salida indicada

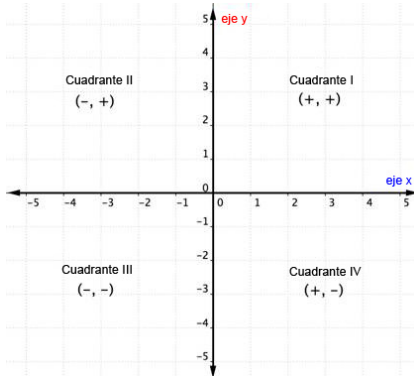
```
class Unidad4{

    public static void main(String[] args) {
        //se indica al constructor columnasxfilas
        PlanoJardin plano=new PlanoJardin(5,4);
        System.out.println("antes de plantar");
        plano.imprimirPlano();
        plano.plantarArbol(0, 3);
        plano.plantarArbol(4,3);
        plano.plantarArbol(2, 0);
        System.out.println("despues de plantar");
        plano.imprimirPlano();
        System.out.println(plano.hayArbol(0,3));
        System.out.println(plano.hayArbol(3,0));
    }
}
```

```
run:
antes de plantar
- - - - -
- - - - -
- - - - -
- - - - -
despues de plantar
A - - - A
- - - - -
- - - - -
- - A - -
true
false
BUILD SUCCESSFUL (total time: 0 seconds)
```

Almacenar un plano de coordenadas de cuatro cuadrantes en una matriz

En el siguiente plano de coordenadas típico de cuatro cuadrantes con 11 posibles valores para el eje x y otros 11 para el eje y por tanto hay un total de 11x11 valores.

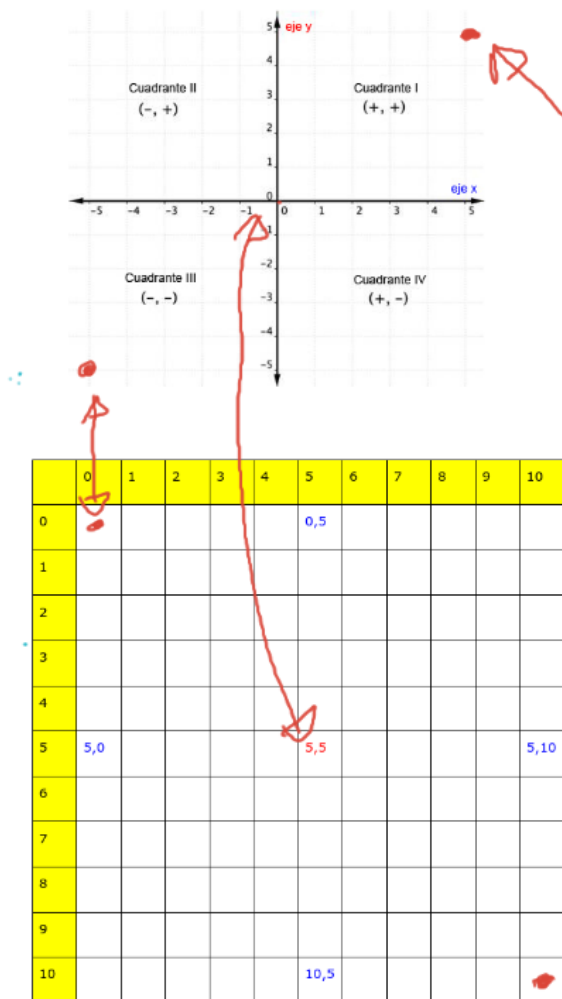


Ahora hay coordenadas negativas. No podemos usar con arrays índices negativos (si en python pero con otro significado). Tengo que trasladar esos valores negativos a valores positivos, por ejemplo a un eje con valores -3,-2,-1,0,1,2,3 le sumo 3 para trabajar con valores positivos en los índices

-3,-2,-1,0,1,2,3

0,1,2,3,4,5,6

Para el plano anterior podemos hacer la siguiente traducción



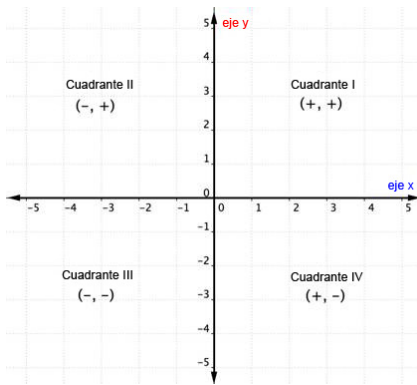
Traducción

x
 \downarrow
 $j = x + 5$

y
 \downarrow
 $i = y + 5$

Ejercicio U4_B4D_E2: Ahora queremos ver el jardín dividido en 4 cuadrantes con una casa C ubicada en el punto central, es decir, punto (0,0). Seguimos considerando que el plano es una cuadrícula de líneas verticales y horizontales y suponemos que ambas son

impares para trabajar con un plano simétrico desde el centro. Igual que en el ejercicio anterior podemos plantar un árbol en cualquier intersección excepto en (0,0) que está la casa.



Escribir de nuevo la clase PlanoJardín de forma que funcione el siguiente main para producir la salida indicada. Al crear el plano especificamos el número de líneas verticales y horizontales en los que lo dividimos. En el centro del plano está de forma fija la casa C, en el resto de los puntos se pueden plantar árboles. Para simplificar, suponemos que las dimensiones del plano son impares.

```
class Unidad4{

    public static void main(String[] args) {
        //se indica al constructor hileras verticales y horizontales
        PlanoJardin plano=new PlanoJardin(11,9);
        System.out.println("Antes de plantar");
        plano.imprimirPlano();
        plano.plantarArbol(2,3);
        plano.plantarArbol(2,-3);
        plano.plantarArbol(-2,3);
        plano.plantarArbol(-2,-3);
        System.out.println("Despues de plantar");
        plano.imprimirPlano();
        System.out.println(plano.hayArbol(2,-3));
        System.out.println(plano.hayArbol(3,0));

    }
}
```

SALIDA:

```

Antes de plantar
- - - - -
- - - - -
- - - - -
- - - - - C - - - - -
- - - - -
- - - - -
- - - - -

Despues de plantar
- - - - -
- - - A - - - A - - -
- - - - -
- - - - - C - - - - -
- - - - -
- - - - -
- - - A - - - A - - -
- - - - -

true
false
BUILD SUCCESSFUL (total time: 0 seconds)

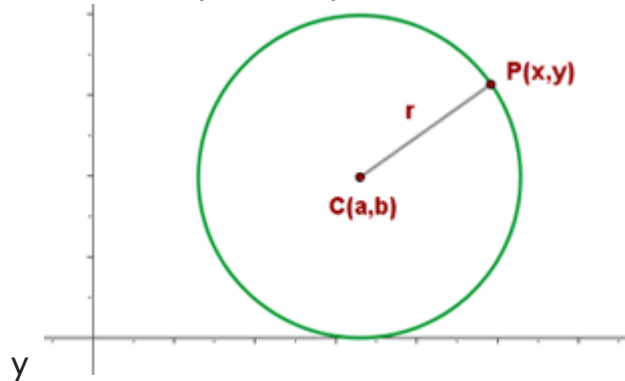
```

REPRESENTAR UNA CIRCUNFERENCIA O UN CÍRCULO EN UNA MATRIZ

Muchos problemas del mundo real quieren resolver cuestiones del tipo “cuántos árboles hay desde el punto (x,y) en un radio de distancia z “. Si representamos la superficie en una matriz tengo que saber “trabajar con círculos dentro de una matriz”.

Ecuación de los puntos que forman una circunferencia y un círculo

Circunferencia es el lugar geométrico de los puntos del plano que equidistan de un punto fijo llamado centro.
 Recuerda que circunferencia y círculo no es lo mismo. Informalmente, la circunferencia se refiere a los puntos de la línea exterior, el círculo incluye también los puntos que encierra dicha línea.



Asociado al concepto de circunferencia/círculo tenemos el concepto de radio. Si C es el centro de un círculo y P cualquier punto P siempre se cumple lo siguiente que la distancia de C a P es un valor r.

$$d(C, P) = r$$

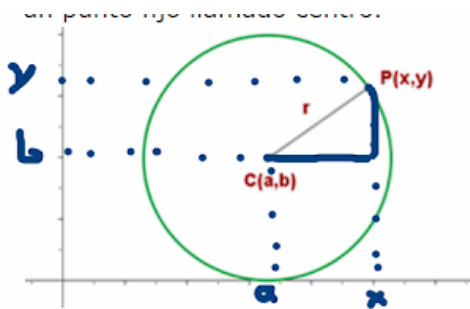
el radio es la distancia de C a P

C y P son puntos con coordenadas del tipo (x,y). supongamos que las coordenadas de C son (a,b) y las de P (x,y) puedo expresar la distancia de C a P de la siguiente manera

$$\sqrt{(x-a)^2 + (y-b)^2} = r$$

O bien Elevando todo al cuadrado obtenemos la ecuación:

$$(x-a)^2 + (y-b)^2 = r^2$$



Esta es la ecuación de la circunferencia, la ecuación del círculo se obtiene simplemente sustituyendo el = por <= para incluir todos los puntos que están "dentro"

$$(x-a)^2 + (y-b)^2 \leq r^2$$

Si la circunferencia tiene como centro el punto (0,0) y radio r la ecuación queda

$$x^2 + y^2 = r^2$$

y por tanto todos los puntos que pertenecen a un **círculo** con centro (0,0) y radio r será

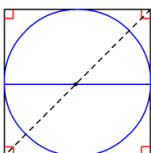
$$x^2 + y^2 \leq r^2$$

Más explicaciones en estos videos (hay muchos similares)

1. Entender la ecuación de distancia de dos puntos en el plano:
<http://www.matematicatuya.com/GRAFICAecuaciones/S1a.html>
2. Entender la ecuación que describe los puntos de una circunferencia
<https://www.youtube.com/watch?v=rkEm5jKj8pY>

representar un círculo en una matriz

Hay muchas posibilidades pero como punto de partida, lo mejor es suponer que tenemos una matriz cuadrada de nxn y que n es impar. Al ser impar podemos deducir un centro para un círculo que ocupa la mayor superficie posible y está dicho círculo repartido de forma simétrica sobre la matriz, es decir, vamos a usar una matriz a modo de cuadrado con un círculo inscrito



En el siguiente ejemplo (sin contar la fila y columna de cabecera, zona amarilla) tenemos una matriz de 11x11 con lo que su centro será la celda 5,5

	0	1	2	3	4	5	6	7	8	9	10
0						0,5					
1											
2											
3											
4											
5	5,0					5,5					5,10
6											
7											
8											
9											
10						10,5					

Para hallar el punto medio: `matriz.length/2=medio`, $11/2=5$. Como la matriz es cuadrada el centro será (5,5)

El radio de la circunferencia más grande posible se visualiza fácilmente ya que la circunferencia circunscrita toca en 4 puntos los límites de la matriz (en azul).

Calcular el valor del radio de la circunferencia inscrita en la matriz

El radio es una distancia o longitud. La unidad de medida pueden ser centímetros, metros o lo que sea. No importa la unidad en este ejercicio, podemos pensar en cualquier unidad de longitud. Lo que me importa ahora es su valor numérico y puedo razonar su valor de muchas maneras. Siempre va a ser con nuestra representación un número entero.

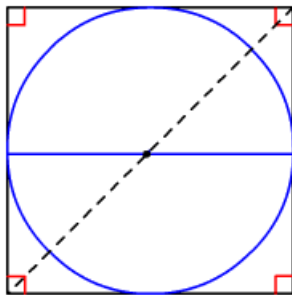
por ejemplo: si el centro es `[cf][cc]`, el radio es la distancia $r=matriz.length-1-cc$ o razonamiento similar que lleva a que el radio sea 5 en la siguiente matriz

RADIO

	0	1	2	3	4	5 cc	6	7	8	9	10 <small>matriz.length-1</small>
--	---	---	---	---	---	---------	---	---	---	---	--------------------------------------

0						0,5					
1											
2											
3											
4											
5	5,0					5,5				5,10	
6											
7											
8											
9											
10						10,5					

Otra forma de verlo más simple para nuestro caso, ya que se trata de un círculo inscrito en un cuadrado, consiste en observar que el diámetro del círculo es igual a la longitud del lado del cuadrado.



como además nuestro cuadrado es una matriz nxn con n impar

diametro= matriz.length

y por tanto

radio=matriz.length/2

“Ver” la circunferencia y círculo inscrito en una matriz.

- Conocemos el centro:[cf][cc]
- Conocemos el radio r: $r = \text{matriz.length} - 1 - cc$ (o calculado de otra manera)
- Por lo tanto, podemos saber si un punto [i][j] pertenece o no a la circunferencia y/o al círculo mayor de la matriz. No podemos trabajar con infinitos puntos, sólo con los puntos que representamos con las combinaciones de los índices de matriz

Es decir si la distancia de (x,y) al centro (a,b) es la misma que la de r es que pertenece a la circunferencia

$$(x - a)^2 + (y - b)^2 = r^2$$

tenemos que adecuar la fórmula a nuestros nombres de variables

```
public class Unidad4{
    public static void main(String[] args) {
        int[][] matriz = new int[11][11];
        int cf = matriz.length / 2; //la coordenada fila del centro
        int cc=cf;//columna centro, en nuestro caso coincide con cf
        int r=matriz.length-1-cc;

        //probamos con NORTE que ya sabemos que pertenece a circunferencia
        int i=0;
        int j=5;
        //true si r^2== (x-a)^2+ (y-b)^2 ver formula dibujo
        //true si r^2== (j-cc)^2+(i-cf)^2
        if(r*r==(j-cc)*(j-cc)+(i-cf)*(i-cf)){
            System.out.println("el punto [0][5] pertenece a la circunferencia");
        }
        i=4;
        j=4;
        if(r*r==(j-cc)*(j-cc)+(i-cf)*(i-cf)){
            System.out.println("el punto [4][4] pertenece a la circunferencia");
        }
    }
}
```

Para saber si pertenece al círculo mayor inscrito simplemente cambiamos el == por <=

```
public class Unidad4{
    public static void main(String[] args) {
        int[][] matriz = new int[11][11];
        int cf = matriz.length / 2; //la coordenada fila del centro
        int cc=cf;//columna centro, en nuestro caso coincide con cf
        int r=matriz.length-1-cc;

        int i=0;
        int j=5;
        //true si r^2== (x-a)^2+ (y-b)^2 ver formula dibujo
        //true si r^2== (j-cc)^2+(i-cf)^2
        if(r*r>=(j-cc)*(j-cc)+(i-cf)*(i-cf)){
            System.out.println("el punto [0][5] pertenece al círculo");
        }
        i=4;
        j=4;
        if(r*r>=(j-cc)*(j-cc)+(i-cf)*(i-cf)){
            System.out.println("el punto [4][4] pertenece al círculo");
        }
    }
}
```

Para ver gráficamente todos los puntos de la matriz que pertenecen a la circunferencia lo que podemos hacer es barrer toda la matriz y ponerle como contenido de celda el valor 1 si el punto pertenece a la circunferencia.

Luego imprimimos la matriz y vemos una suerte de circunferencia, una circunferencia tiene infinitos puntos y representando tan pocos puntos se ve un perfil muy abrupto.

```
public class Unidad4{
    public static void main(String[] args) {
        int[][] matriz = new int[11][11];
        int cf = matriz.length / 2; //la coordenada fila del centro
        int cc=cf;//columna centro, en nuestro caso coincide con cf
        int r=matriz.length-1-cc;

        //marcamos la circunferencia con 1s para imprimir luego
        for (int i = 0; i < matriz.length; i++) {
            int yy=(i-cf)*(i-cf);//el (y-b)^2 de la fórmula
            for (int j = 0; j < matriz[i].length; j++) {
                int xx=(j-cc)*(j-cc);//el (x-a)^2 de la fórmula
                if(Math.sqrt(xx+yy)==r){//la misma fórmula pero ahora con sqrt
                    matriz[i][j]=1;
                }
            }
        }
    }
}
```

```

    }
}

//ahora imprimir matriz. No era realmente necesario almacenar los 1
//podimos haberlos calculado aquí "al vuelo"
for (int i = 0; i < matriz.length; i++) {
    for (int j = 0; j < matriz[i].length; j++) {
        System.out.print(matriz[i][j]+" ");
    }
    System.out.println("");
}
}
}
}

```

Representar un círculo en una matriz

si convertimos la ecuación en una inecuación con \leq , en lugar de poner solamente a 1 los puntos de la circunferencia ponemos también a 1 los que están dentro, es decir, todo el círculo

```

if(Math.sqrt(xx+yy)<=r){
    matriz[i][j]=1;
}

```

Con este pequeño cambio se imprime una suerte de "círculo"

```

0 0 0 0 0 1 0 0 0 0 0
0 0 1 1 1 1 1 1 1 0 0
0 1 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1 1
0 1 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1 0
0 0 1 1 1 1 1 1 1 0 0
0 0 0 0 0 1 0 0 0 0 0

```

No es posible dibujar un buen círculo manejando pocos puntos, observa por ejemplo "el salto" de [5][0] a [4][1]. No tengo un posible valor intermedio para tener un borde más suave ya que [4][0] no pertenece al círculo según la ecuación, por ejemplo, si hago cuentas el punto [4][0.5] también pertenece al círculo según ecuación, pero este punto no lo puedo representar asociado a índices de matriz que tiene que ser enteros. Por eso vemos un círculo poco suave

Ejercicio U4_B4D_E3: bocata de calamares en coderunner