

Practica N1

Problemas asignados para entrega

1. Algebra: ecuación lineal 2 x 2. Diseñe la clase EcuacionLineal para un sistema de ecuaciones de 2 x 2.

$$\begin{aligned} ax + by &= e \\ cx + dy &= f \\ x &= \frac{ed - bf}{ad - bc} \quad y = \frac{af - ec}{ad - bc} \end{aligned}$$

La clase contiene:

- a) Los atributos privados a, b, c, d y f.
- b) Un constructor para los argumentos para a, b, c, d y f.
- c) Un método tieneSolucion() que devuelve true si $ad - bc$ no es cero.
- d) Dos métodos getX() y getY() que retornan la solución de la ecuación.

Dibuje el diagrama UML de la clase e implemente. Escriba un programa de prueba (Test) que solicite al usuario ingresar a, b, c, d, e y f y muestre el resultado. Si $ad - bc$ es 0, indique que "La ecuación no tiene solución".

Ejemplo de entrada

Ingrese a, b, c, d, e, f: 9.0 4.0 3.0 -5.0 -6.0 -21.0

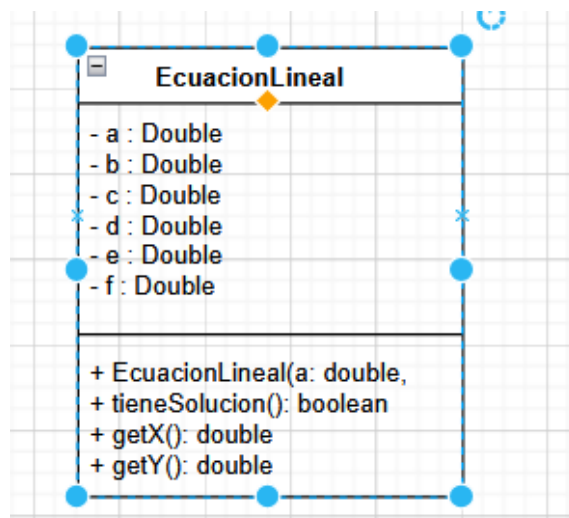
Ingrese a, b, c, d, e, f: 1.0 2.0 2.0 4.0 4.0 5.0

Ejemplo de salida

x = -2.0, y = 3.0

La ecuación no tiene solución

diagrama



J EcuacionLineal.java •
ejercicio1 > J EcuacionLineal.java > EcuacionLineal > EcuacionLineal(double, double, double, double, double, double)

```
1
2 package ejercicio1;
3
4 public class EcuacionLineal {
5     private double a;
6     private double b;
7     private double c;
8     private double d;
9     private double e;
10    private double f;
11
12    public EcuacionLineal(double a, double b, double c,
13        double d, double e, double f) {
14        this.a = a;
15        this.b = b;
16        this.c = c;
17        this.d = d;
18        this.e = e;
19        this.f = f;
20    }
21
22    public boolean tieneSolucion() {
23        return (a * d - b * c) != 0;
24    }
25
26    public double getX() {
27        return (e * d - b * f) / (a * d - b * c);
28    }
29
30    public double getY() {
31        return (a * f - e * c) / (a * d - b * c);
32    }
33 }
```

Resultado:

```
PS E:\UMSA\third\121>
PS E:\UMSA\third\121> e.; cd 'e:\UMSA\third\121\practica
ge\529595223cf2992e0306755342991ad6\redhat.java\jdt_ws\pr
Ingreso a: 9.0
Ingreso b: 4.0
Ingreso c: 3.0
Ingreso d: -5.0
Ingreso e: -6.0
Ingreso f: -21.0
x = -2.0, y = 3.0
PS E:\UMSA\third\121\practica1>
```

EcuacionLineal.java • TestEcuacionLineal.java
ejercicio1 > J TestEcuacionLineal.java > TestEcuacionLineal

```
1 package ejercicio1;
2 import java.util.Scanner;
3
4 public class TestEcuacionLineal {
5     Run | Debug
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner(System.in);
8
9         System.out.print(s:"Ingreso a: ");
10        double a = scanner.nextDouble();
11        System.out.print(s:"Ingreso b: ");
12        double b = scanner.nextDouble();
13        System.out.print(s:"Ingreso c: ");
14        double c = scanner.nextDouble();
15        System.out.print(s:"Ingreso d: ");
16        double d = scanner.nextDouble();
17        System.out.print(s:"Ingreso e: ");
18        double e = scanner.nextDouble();
19        System.out.print(s:"Ingreso f: ");
20        double f = scanner.nextDouble();
21
22        EcuacionLineal ecuacion = new EcuacionLineal(a, b, c, d, e, f);
23
24        if (ecuacion.tieneSolucion()) {
25            System.out.println("x = " + ecuacion.getX() + ", y = " + ecuacion.getY());
26        } else {
27            System.out.println(x:"La ecuación no tiene solución");
28        }
29
30        scanner.close();
31    }
32 }
```

2 Algebra: Ecuaciones Cuadráticas, Diseñe la clase EcuacionLineal para la ecuación cuadrática

$ax^2 + bx + c = 0$. La clase contiene:

- a) Los atributos privados a, b y c, que representan a los 3 coeficientes.
- b) Un constructor para los argumentos para a, b y c.
- c) Un método getDiscriminante() que devuelve el discriminante $b^2 - 4ac$.
- d) Dos métodos getRaiz1() y getRaiz2() que retornan las dos raíces de la ecuación.

Estos $r1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$ y $r2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$ métodos solo son útiles si el discriminante es positivo. Si el discriminante es negativo, devuelva 0.

Dibuje el diagrama UML de la clase e implemente. Escriba un programa de prueba (Test) que solicite al usuario introducir los valores de a, b y c, y muestre el resultado según el discriminante. Si el discriminante es positivo, muestre las dos raíces. Si el discriminante es 0, muestre la única raíz. De lo contrario, muestre "La ecuación no tiene raíces reales".

Ejemplo de entrada

Ingrese a, b, c: 1.0 3 1

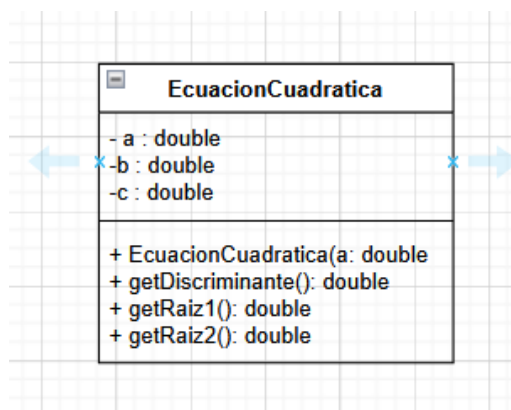
Ejemplo de salida

La ecuación tiene dos raíces -0.381966 y -2.61803

La ecuación tiene una raíz -1

La ecuación no tiene raíces reales

Diagrama



cio2 > J EcuacionCuadratica.java > EcuacionCuadratica

```
1 package ejercicio2;
2
3 public class EcuacionCuadratica {
4     private double a;
5     private double b;
6     private double c;
7
8     public EcuacionCuadratica(double a, double b, double c) {
9         this.a = a;
10        this.b = b;
11        this.c = c;
12    }
13
14    public double getDiscriminante() {
15        return b * b - 4 * a * c;
16    }
17
18    public double getRaiz1() {
19        double discriminante = getDiscriminante();
20        if (discriminante < 0) {
21            return 0;
22        }
23        return (-b + Math.sqrt(discriminante)) / (2 * a);
24    }
25
26    public double getRaiz2() {
27        double discriminante = getDiscriminante();
28        if (discriminante < 0) {
29            return 0;
30        }
31        return (-b - Math.sqrt(discriminante)) / (2 * a);
32    }
33 }
```

Resultado

```
PS E:\UMSA\third\121\practical1>
PS E:\UMSA\third\121\practical1> e.; cd 'e:\UMSA\third\121\practical1';
-cp 'C:\Users\heicar_23\AppData\Roaming\Code\User\workspaceStorage\52
```

```
Ingrese a: 1.0
Ingresa b: 3
Ingresa 1c: 1
La ecuación tiene dos raíces -0.3819660112501051 y -2.618033988749895
PS E:\UMSA\third\121\practical1>
```

cio2 > J TestEcuacionCuadratica.java > TestEcuacionCuadratica > main(String[])

```
1 package ejercicio2;
2 import java.util.Scanner;
3
4 public class TestEcuacionCuadratica {
5     Run | Debug
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner(System.in);
8
9         System.out.print(s:"Ingresa a: ");
10        double a = scanner.nextDouble();
11        System.out.print(s:"Ingresa b: ");
12        double b = scanner.nextDouble();
13        System.out.print(s:"Ingresa 1c: ");
14        double c = scanner.nextDouble();
15
16        EcuacionCuadratica ecuacion = new EcuacionCuadratica(a, b, c);
17        double discriminante = ecuacion.getDiscriminante();
18
19        if (discriminante > 0) {
20            double raiz1 = ecuacion.getRaiz1();
21            double raiz2 = ecuacion.getRaiz2();
22            System.out.println("La ecuación tiene dos raíces " + raiz1 + " y " + raiz2);
23        } else if (discriminante == 0) {
24            double raiz = ecuacion.getRaiz1();
25            System.out.println("La ecuación tiene una raíz " + raiz);
26        } else {
27            System.out.println(x:"La ecuación no tiene raíces reales");
28        }
29
30        scanner.close();
31    }
32 }
```

3. Estadísticas: calcular el promedio y la desviación estándar. En las aplicaciones comerciales, a menudo se le pide que calcule el promedio y la desviación estándar de los datos. La desviación estándar es una estadística que le indica cuán estrechamente se agrupan todos los diversos datos alrededor de la media en un conjunto de datos. Por ejemplo, ¿cuál es la edad promedio de los estudiantes en una clase? ¿Qué tan cerca están las edades? Si todos los estudiantes tienen la misma edad, la desviación es 0. Utilice la siguiente fórmula:

$$promedio = \frac{\sum_{i=1}^n x_i}{n} \quad desviacion = \sqrt{\frac{\sum_{i=1}^n (x_i - promedio)^2}{n - 1}}$$

Su programa debe contener los siguientes métodos:

- promedio() obtiene el promedio de los valores en punto flotante
- desviacion() devuelve la desviación estándar de los valores en punto flotante.

Escriba un programa que solicite al usuario ingresar 10 números, muestre el promedio y las desviaciones estándar de estos números.

****Ejemplo de entrada****

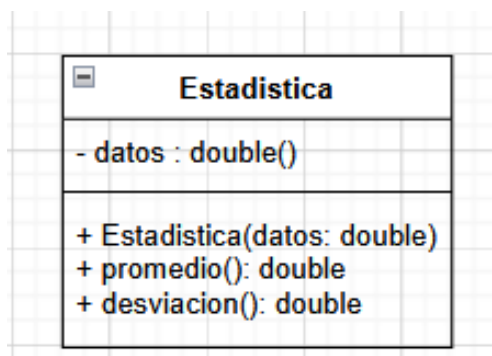
1.9 2.5 3.7 2 1 6 3 4 5 2

****Ejemplo de salida****

El promedio es 3.11

La desviación estándar es 1.55738

Obtenga la solución en la Programación Modular-Estructurada y luego en Programación Orientada a Objetos, para ello dibuje el diagrama UML para la clase Estadística y luego implemente la clase. Agregue de forma escrita las ventajas de utilizar la P.O.O.



ejercicio3 > J Estadistica.java > Estadistica

```
1 package ejercicio3;
2
3 public class Estadistica {
4     private double[] datos;
5
6     public Estadistica(double[] datos) {
7         this.datos = datos;
8     }
9
10    public double promedio() {
11        double suma = 0;
12        for (double num : datos) {
13            suma += num;
14        }
15        return suma / datos.length;
16    }
17
18    public double desviacion() {
19        double prom = promedio();
20        double sumaCuadrados = 0;
21        for (double num : datos) {
22            sumaCuadrados += Math.pow(num - prom, 2);
23        }
24        return Math.sqrt(sumaCuadrados / (datos.length - 1));
25    }
26 }
```

cicio3 > J TestEstadistica.java > TestEstadistica

```
1 package ejercicio3;
2
3 import java.util.Scanner;
4
5 public class TestEstadistica {
6     Run | Debug
7     public static void main(String[] args) {
8         Scanner scanner = new Scanner(System.in);
9         double[] numeros = new double[10];
10
11         System.out.print(s:"Ingrese 10 números separados por espacios: ");
12         for (int i = 0; i < 10; i++) {
13             numeros[i] = scanner.nextDouble();
14         }
15
16         Estadistica stats = new Estadistica(numeros);
17         double prom = stats.promedio();
18         double dev = stats.desviacion();
19
20         System.out.println("El promedio es " + String.format(format:"%.2f", prom));
21         System.out.println("La desviación estándar es " + String.format(format:"%.5f", dev));
22
23         scanner.close();
24     }
25 }
```

```
PS E:\UMSA\third\121\practica1> e;; cd "E:\UMSA\third\121\practica1";
-cp "C:\Users\heicar_23\AppData\Roaming\Code\User\workspaceStorage\529
Ingrese 10 números separados por espacios: 1.9 2.5 3.7 2 1 6 3 4 5 2
El promedio es 3.11
La desviación estándar es 1.55738
PS E:\UMSA\third\121\practica1>
```