

PROGRAMACIÓN (EUI). Curso 2001-2002
*Práctica 2. RESOLUCIÓN DE UNA ECUACIÓN DE
SEGUNDO GRADO*

F. Marqués y N. Prieto

Índice General

1	Introducción. Algoritmos numéricos	1
2	El problema	2
2.1	Solución de la ecuación	2
3	El programa	2
3.1	Estrategia de resolución	3
3.2	Variables y tipos de datos	3
3.3	Impresión de resultados	3
4	Introducción de los valores de los coeficientes	4
4.1	Argumentos de un programa	4
4.2	Argumentos numéricos	4

1 Introducción. Algoritmos numéricos

Uno de los primeros usos de los computadores fue su utilización en la resolución de problemas de índole numérica y estadística: tabulación de funciones elementales, determinación de tablas de trayectorias de proyectiles, resolución de ecuaciones, etc.

Los algoritmos basados en el uso de las operaciones aritméticas elementales se conocen con el nombre de *algoritmos numéricos*. Estos juegan un papel importante en las matemáticas debido a que muchas operaciones pueden reducirse a otras más elementales, habitualmente mediante un uso reiterado de las últimas.

Los algoritmos numéricos se dan usualmente en forma de instrucciones verbales o diversas clases de fórmulas y esquemas. Algoritmos numéricos sobradamente conocidos son los que permiten resolver una ecuación de segundo grado, un sistema lineal de ecuaciones, o extraer una raíz cuadrada. La utilización de computadoras en la resolución de problemas que implican este tipo de algoritmos permite ahorrar al hombre una gran cantidad de trabajo tedioso y rutinario.

En esta práctica se plantea el problema de calcular de forma general mediante un programa, la solución o soluciones de una ecuación de segundo grado de coeficientes reales cualesquiera. El método que se seguirá consistirá en la aplicación, casi inmediata, de la conocida fórmula de resolución de ecuaciones de dicho tipo.

2 El problema

Se desea resolver de forma general, la siguiente ecuación de segundo grado:

$$ax^2 + bx + c = 0$$

donde, a , b y c , que son los coeficientes de la ecuación, son números reales cualesquiera, y x , la variable, cuyo valor o valores se desea conocer, puede ser un número real o complejo.

2.1 Solución de la ecuación

La solución de la ecuación anterior depende de los valores de los coeficientes y, en función de los mismos, cabe distinguir entre los siguientes casos:

- si a , b y c valen simultáneamente 0, entonces cualquier número es solución de la ecuación
- si a y b son simultáneamente 0, pero c no lo es, la ecuación es incorrecta
- si a vale 0 y b es distinto de 0, entonces la ecuación es de primer grado, y la solución x es el número real $-c/b$
- si a es distinto de 0, entonces la solución de la ecuación se puede obtener siempre aplicando la fórmula:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

donde al valor $(b^2 - 4ac)$ se denomina *discriminante*. En dicho caso:

- si el *discriminante* es 0 la solución es única y es un número real,
- si el *discriminante* es positivo las dos soluciones son números reales,
- si el *discriminante* es negativo las dos soluciones son números complejos.

3 El programa

Es necesario, para poder resolver el problema mediante un programa determinar tanto la estrategia, o pasos que seguirá el mismo en la solución del problema, como las características de los datos que intervengan en el mismo. Todo ello para expresar mediante el lenguaje de programación deseado la estructura del programa solución, y las variables y tipos de los datos implicados.

3.1 Estrategia de resolución

Para resolver el problema, de forma general, mediante un programa, se puede seguir una estrategia como la siguiente:

1. Pedir inicialmente los valores de los coeficientes al usuario,
2. En función de ellos decidir si la ecuación:
 - Es incorrecta,
 - tiene un número infinito de soluciones,
 - es una ecuación de primer grado, con solución real, o
 - es una ecuación de segundo grado y:
 - el *discriminante* vale 0 y la ecuación tiene una única solución real
 - el *discriminante* es positivo, y la ecuación tiene soluciones reales, o
 - el *discriminante* es negativo y la ecuación tiene soluciones complejas,
3. escribir en la salida la advertencia o solución hallada según corresponda.

3.2 Variables y tipos de datos

Tanto los coeficientes, a , b y c , como otros posibles números, como el **discriminante**, son números reales, por lo que se mantendrán a lo largo de la ejecución mediante variables de dicho tipo, por ejemplo se pueden declarar como variables de tipo **double**.

Como se ha visto, según el tipo de ecuación, algunas soluciones pueden ser números complejos. Sin embargo, en el lenguaje no existe el tipo de datos número complejo de forma predefinida, por lo que se hace necesario representar dichos números de alguna forma alternativa. Se puede utilizar, por su sencillez, la forma cartesiana, en la que cada número complejo se representa mediante un par de valores de tipo real (**double**), uno para la parte real, y otro para la imaginaria. De esta manera, si n es un número real negativo cualquiera, entonces el valor complejo \sqrt{n} , es equivalente a $\sqrt{|n|}i$.

Por ejemplo, el valor complejo $\sqrt{-16}$, es equivalente a $\sqrt{|-16|}i$, esto es: $4i$.

3.3 Impresión de resultados

La impresión de los resultados debe realizarse siempre de forma que sea legible y no de lugar a ambigüedades. Para este programa, además, deberán tenerse en cuenta las siguientes consideraciones:

- Si existe una única solución de la ecuación, como por ejemplo cuando el *discriminante* vale 0, ésta no deberá aparecer duplicada.
- Cuando se escriba un número complejo, deberá quedar claramente diferenciada su parte real y su parte imaginaria, como por ejemplo:

Soluciones: (3.2145 + 14.48i) (3.2145 - 14.48i)

4 Introducción de los valores de los coeficientes

Para la introducción de los coeficientes se utilizarán dos métodos distintos. El primero de ellos, que no merece consideración especial, ya que ya ha sido utilizado, consiste en hacer uso de las primitivas definidas en el *paquete nsIO* para la definición de ficheros de entrada/salida e introducción de datos.

La segunda posibilidad, que se estudiará a continuación, consiste en introducir los datos como *argumentos del programa* que se va a ejecutar.

4.1 Argumentos de un programa

Al estudiar la estructura de un programa en el tema 2º, se ha visto que ésta era de la forma:

```
class NombrePrograma {
    public static void main (String args[]) {
        ...
    }
}
```

donde el método `main` indica el segmento de código que se ejecutará inicialmente. Dicho método tiene un argumento declarado: `String args[]`, que hace referencia a un grupo de variables enumeradas (un `array` en términos del Java) de tipo `String`.

Cuando el usuario ejecuta el programa `NombrePrograma`, puede hacerlo utilizando argumentos, como se hace en el siguiente ejemplo en el que se utilizan tres argumentos, uno de ellos el número 37:

```
java NombrePrograma ejem1 37 ejem3
```

Cuando el programa `NombrePrograma` se ejecuta puede hacer referencia a los argumentos con los que arranca, mediante las variables: `args[0]`, cadena de caracteres (`String`) de valor "ejem1"; `args[1]`, cadena de caracteres (`String`) de valor "37" y `args[2]`, cadena de caracteres (`String`) de valor "ejem3"

Nótese que los argumentos:

- pueden ser tantos como se desee,
- se separan entre si por espacios en blanco, y
- son todos de tipo `String`

4.2 Argumentos numéricos

Si se desea hacer uso de argumentos numéricos, entonces es necesario transformar a número la `String` con que se reconocen dentro del programa los argumentos con que se inicia el mismo.

Para hacer esto posible, el lenguaje Java hace uso de unos tipos de datos numéricos especiales, denominados *tipos de envoltura*, que se estudiarán más adelante, y que tienen operaciones específicas para realizar dicho tipo de transformaciones. Los tipos `Int` y `Double` son, por ejemplo, *tipos de envoltura* de los tipos `int` y `double`, respectivamente.

El siguiente ejemplo, donde se desea introducir un valor entero y otro real en la línea de comando del programa, muestra cómo puede realizarse todo esto en Java. Para ello se leen dos argumentos numéricos como `String`, almacenadas en `argv[0]` y `argv[1]`, y a continuación se transforman, respectivamente, en variables de tipo `int` y `double`, escribiéndose dichos números seguidamente:

```
class ejem2 {
static public void main(String argv[]) {

    int i = new Integer(argv[0]).intValue();
    double d = new Double(argv[1]).doubleValue();

    System.out.println("El primer numero es: " + i);
    System.out.println("El segundo numero es: " + d);

}
} /* fin de ejem2 */
```

Donde los métodos `Integer()` y `Double()`, permiten construir objetos de dicho tipo, (`Integer` y `Double`), a partir de valores de tipo `String`, y los métodos `intValue()` y `doubleValue()` devuelven, respectivamente, un valor elemental de tipo `int` y `double`.

Así, con ello, es posible efectuar una llamada inicial al programa como la siguiente:

```
java ejem2 12 123.678
```

Es importante señalar que se producirá un error de ejecución (una excepción) en el caso de que alguno de los argumentos no tenga las características previstas, o si faltara alguno de ellos. Pruébese, por ejemplo, a introducir un carácter donde se prevee un número.