

## Ejercicio 1.11

1.11. Un economista te ha encargado un programa para realizar cálculos con el IVA. La aplicación debe solicitar la base imponible y el IVA que se debe aplicar. Muestra en pantalla el importe correspondiente al IVA y al total.

```
Start Page x Ejercicio1_IVA.java x
Source History

4
5 package com.mycompany.ejerciciol_iva;
6
7 import java.util.Scanner;
8
9 /**
10  * @author Abraham
11  */
12 public class Ejerciciol_IVA {
13
14     public static void main(String[] args) {
15         Scanner scanner = new Scanner(System.in);
16
17         // Solicitar el importe al usuario
18         System.out.print("Ingrese base imponible: ");
19         double importe = scanner.nextDouble();
20
21         // Solicitar el porcentaje de impuesto al usuario
22         System.out.print("Ingrese el iva a aplicar: ");
23         double porcentajeImpuesto = scanner.nextDouble();
24
25         // Calcular el impuesto y el total
26         double impuesto = (porcentajeImpuesto / 100) * importe;
27         double total = importe + impuesto;
28
29         // Mostrar los resultados
30         System.out.println("Impuesto: " + impuesto);
31         System.out.println("Total: " + total);
32
33         // Cerrar el objeto Scanner para evitar fugas de recursos
34         scanner.close();
35     }
36 }
37
```

```
Output - Run (ejercicio1_IVA) x
cd C:\Users\Abraham\Documents\NetBeansProjects\ejerciciol_IVA; "JAVA_HOME=C:\Program Files\Java\jdk-18.0.2" cmd /c "%C:\Progr
Scanning for projects...

-----< com.mycompany:ejerciciol_IVA >-----
Building ejerciciol_IVA 1.0-SNAPSHOT
from pom.xml
-----[ jar ]-----

--- resources:3.3.1:resources (default-resources) @ ejerciciol_IVA ---
skip non existing resourceDirectory C:\Users\Abraham\Documents\NetBeansProjects\ejerciciol_IVA\src/main/resources

--- compiler:3.11.0:compile (default-compile) @ ejerciciol_IVA ---
Nothing to compile - all classes are up to date

--- exec:3.1.0:exec (default-cli) @ ejerciciol_IVA ---
Ingrese base imponible: 10
Ingrese el iva a aplicar: 21
Impuesto: 2.1
Total: 12.1

BUILD SUCCESS

Total time: 12.024 s
Finished at: 2023-10-17T19:09:52+01:00
-----
```

## Ejercicio 1.12

- 1.12** Escribe un programa que tome como entrada un número entero e indique qué cantidad hay que sumarle para que el resultado sea múltiplo de 7. Un ejemplo:
- A 2 hay que sumarle 5 para que el resultado ( $2 + 5 = 7$ ) sea múltiplo de 7.
  - A 13 hay que sumarle 1 para que el resultado ( $13 + 1 = 14$ ) sea múltiplo de 7.
- Si proporcionas el número 2 o el 13, la salida de la aplicación debe ser 5 o 1, respectivamente.
- Pista: El operador módulo puede ser muy útil para solucionar esta actividad.

```
Source History
1
2 package com.mycompany.ejercicio2;
3
4 import java.util.Scanner;
5
6
7 public class Ejercicio2 {
8
9     public static void main(String[] args) {
10         // Crear un objeto Scanner para leer la entrada del usuario
11         Scanner scanner = new Scanner(System.in);
12
13         // Solicitar un número entero al usuario
14         System.out.print("Ingrese un número entero: ");
15         int numero = scanner.nextInt();
16
17         // Calcular la cantidad que se debe sumar para que sea múltiplo de 7
18         int residuo = numero % 7;
19         int cantidadSumar = (residuo == 0) ? 0 : 7 - residuo;
20
21         // Calcular el nuevo número
22         int nuevoNumero = numero + cantidadSumar;
23
24         // Mostrar los resultados
25         System.out.println("Cantidad que hay que sumar: " + cantidadSumar);
26         System.out.println("Nuevo número para que sea múltiplo de 7: " + nuevoNumero);
27
28         // Cerrar el objeto Scanner para evitar fugas de recursos
29         scanner.close();
30     }
31 }
32
33
34
```

```
Output - Run (Ejercicio2) x
cd C:\Users\Abraham\Documents\NetBeansProjects\Ejercicio2; "JAVA_HOME=C:\\Program Files\\Java\\jdk-18.0.2" cmd
Scanning for projects...

-----< com.mycompany:Ejercicio2 >-----
Building Ejercicio2 1.0-SNAPSHOT
from pom.xml
-----[ jar ]-----

--- resources:3.3.1:resources (default-resources) @ Ejercicio2 ---
- skip non existing resourceDirectory C:\Users\Abraham\Documents\NetBeansProjects\Ejercicio2\src\main\resources

--- compiler:3.11.0:compile (default-compile) @ Ejercicio2 ---
Changes detected - recompiling the module! :source
Compiling 1 source file with javac [debug target 18] to target\classes

--- exec:3.1.0:exec (default-cli) @ Ejercicio2 ---
Ingrese un número entero: 25
Cantidad que hay que sumar: 3
Nuevo número para que sea múltiplo de 7: 28

BUILD SUCCESS

Total time: 9.969 s
Finished at: 2023-10-17T19:24:15+01:00
```

## Ejercicio 1.13

**1.13.** Modifica la Actividad de Aplicación 1.12 para que, indicando dos números  $n$  y  $m$ , diga qué cantidad hay que sumarle a  $n$  para que sea múltiplo de  $m$ .

The screenshot displays an IDE with two panels. The top panel shows the source code for a Java class named `Ejercicio3`. The code uses `Scanner` to take two integers as input, calculates the remainder of the first number divided by the second, and determines the amount needed to reach the next multiple. The bottom panel shows the output of running the program, including build logs and the interactive session where the user enters 30 and 18, resulting in a sum of 6 to reach the next multiple of 18, which is 36.

```
1 package com.mycompany.ejercicio3;
2
3 import java.util.Scanner;
4
5 /**
6  *
7  * @author Abraham
8  */
9 public class Ejercicio3 {
10
11     public static void main(String[] args) {
12         // Crear un objeto Scanner para leer la entrada del usuario
13         Scanner scanner = new Scanner(System.in);
14
15         // Solicitar un número entero al usuario
16         System.out.print("Ingrese un número entero: ");
17         int numero = scanner.nextInt();
18         // Solicitar otro número entero al usuario
19         System.out.print("Ingrese otro número entero para calcular el múltiplo: ");
20         int numeroMultiplo = scanner.nextInt();
21
22         // Calcular la cantidad que se debe sumar a n para que sea múltiplo de m
23         int residuo = numero % numeroMultiplo;
24         int cantidadSumar = (residuo == 0) ? 0 : numeroMultiplo - residuo;
25
26         // Calcular el nuevo número
27         int nuevoNumero = numero + cantidadSumar;
28
29         // Mostrar los resultados
30         System.out.println("Cantidad que hay que sumar: " + cantidadSumar);
31         System.out.println("Nuevo número para que sea múltiplo de " +
32             numeroMultiplo + ": " + nuevoNumero);
33
34         // Cerrar el objeto Scanner para evitar fugas de recursos
35         scanner.close();
36     }
37 }
38
39
```

Output - Run (Ejercicio3) x

```
cd C:\Users\Abraham\Documents\NetBeansProjects\Ejercicio3; "JAVA_HOME=C:\\Program Files\\Java\\jdk-18.0.2" cmd /c "%C:\\
Scanning for projects...

-----< com.mycompany:Ejercicio3 >-----
Building Ejercicio3 1.0-SNAPSHOT
from pom.xml
-----[ jar ]-----

--- resources:3.3.1:resources (default-resources) @ Ejercicio3 ---
skip non existing resourceDirectory C:\Users\Abraham\Documents\NetBeansProjects\Ejercicio3\src\main\resources

--- compiler:3.11.0:compile (default-compile) @ Ejercicio3 ---
Changes detected - recompiling the module! :source
Compiling 1 source file with javac [debug target 18] to target\classes

--- exec:3.1.0:exec (default-cli) @ Ejercicio3 ---
Ingrese un número entero: 30
Ingrese otro número entero para calcular el múltiplo: 18
Cantidad que hay que sumar: 6
Nuevo número para que sea múltiplo de 18: 36

BUILD SUCCESS

Total time: 47.522 s
Finished at: 2023-10-17T19:36:14+01:00
-----
```



## Ejercicio 1.14

1.14. Crea un programa que pida la base y la altura de un triángulo y muestre su área.

$$\text{Área triángulo} = \frac{\text{base} \cdot \text{altura}}{2}$$

```
Source History
1
2 package com.mycompany.ejercicio4;
3
4 import java.util.Scanner;
5
6
7 public class Ejercicio4 {
8
9     public static void main(String[] args) {
10         Scanner scanner = new Scanner(System.in);
11
12         // Solicitar base
13         System.out.print("Ingrese la base del triángulo: ");
14         int base = scanner.nextInt();
15
16         // Solicitar altura
17         System.out.print("Ingrese la altura del triángulo: ");
18         int altura = scanner.nextInt();
19
20         //calcular área
21         double area = (double)base * altura / 2;
22
23         //mostrar el área
24         System.out.println("El área del triángulo es: " + area);
25     }
26 }
27
```

```
Output - Run (Ejercicio4) x
cd C:\Users\Abraham\Documents\NetBeansProjects\Ejercicio4; "JAVA_HOME=C:\Program Files\Java\jdk-18.0.2" cmd /c
Scanning for projects...
-----< com.mycompany:Ejercicio4 >-----
Building Ejercicio4 1.0-SNAPSHOT
from pom.xml
-----[ jar ]-----
--- resources:3.3.1:resources (default-resources) @ Ejercicio4 ---
skip non existing resourceDirectory C:\Users\Abraham\Documents\NetBeansProjects\Ejercicio4\src\main\resources
--- compiler:3.11.0:compile (default-compile) @ Ejercicio4 ---
Changes detected - recompiling the module! :source
Compiling 1 source file with javac [debug target 18] to target\classes
--- exec:3.1.0:exec (default-cli) @ Ejercicio4 ---
Ingrese la base del triángulo: 7
Ingrese la altura del triángulo: 5
El área del triángulo es: 17.5
BUILD SUCCESS
Total time: 9.482 s
Finished at: 2023-10-17T19:45:16+01:00
```

## Ejercicio1.15

1.15. Dado el siguiente polinomio de segundo grado:

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

crea un programa que pida los coeficientes  $a$ ,  $b$  y  $c$ , así como el valor de  $x$ , y calcule el valor correspondiente de  $y$ .

```
Source History [Icons]
1
2 package com.mycompany.ejercicio5;
3
4 import java.util.Scanner;
5
6 public class Ejercicio5 {
7
8     public static void main(String[] args) {
9         Scanner scanner = new Scanner(System.in);
10
11         // Solicitar los coeficientes a, b, c y el valor de x al usuario
12         System.out.print("Ingrese el coeficiente a: ");
13         double a = scanner.nextDouble();
14
15         System.out.print("Ingrese el coeficiente b: ");
16         double b = scanner.nextDouble();
17
18         System.out.print("Ingrese el coeficiente c: ");
19         double c = scanner.nextDouble();
20
21         System.out.print("Ingrese el valor de x: ");
22         double x = scanner.nextDouble();
23
24         // Calcular el valor de y para el polinomio de segundo grado
25         double y = calcularPolinomio(a, b, c, x);
26
27         // Mostrar el resultado
28         System.out.println("El valor de y es: " + y);
29
30         // Cerrar el objeto Scanner para evitar fugas de recursos
31         scanner.close();
32     }
33
34     // Método para calcular el valor de y en un polinomio de segundo grado
35     private static double calcularPolinomio(double a, double b, double c, double x) {
36         return (a * x * x) + (b * x) + c;
37     }
38 }
39
```

Output - Run (Ejercicio5) x

```
cd C:\Users\Abraham\Documents\NetBeansProjects\Ejercicio5; "JAVA_HOME=C:\\Program Files\\Java\\jdk-18.0.2" cmd /c "%C:\\Pro
Scanning for projects...

-----< com.mycompany:Ejercicio5 >-----
Building Ejercicio5 1.0-SNAPSHOT
  from pom.xml
-----[ jar ]-----

--- resources:3.3.1:resources (default-resources) @ Ejercicio5 ---
skip non existing resourceDirectory C:\Users\Abraham\Documents\NetBeansProjects\Ejercicio5\src\main\resources

--- compiler:3.11.0:compile (default-compile) @ Ejercicio5 ---
Nothing to compile - all classes are up to date


--- exec:3.1.0:exec (default-cli) @ Ejercicio5 ---
Ingrese el coeficiente a: 5
Ingrese el coeficiente b: 3
Ingrese el coeficiente c: 2
Ingrese el valor de x: 4
El valor de y es: 94.0

-----
BUILD SUCCESS
-----

Total time: 9.381 s
Finished at: 2023-10-17T19:54:04+01:00
-----
```

## Ejercicio1.16

**1.16.** Diseña una aplicación que solicite al usuario que introduzca una cantidad de segundos. La aplicación debe mostrar cuántas horas, minutos y segundos hay en el número de segundos introducidos por el usuario.

```
Source History 
4 import java.util.Scanner;
5
6 public class Ejercicio6 {
7
8     public static void main(String[] args) {
9         // Crear un objeto Scanner para leer la entrada del usuario
10        Scanner scanner = new Scanner(System.in);
11
12        // Solicitar al usuario que ingrese la cantidad de segundos
13        System.out.print("Ingrese la cantidad de segundos: ");
14        int segundosTotales = scanner.nextInt();
15
16        // Calcular horas, minutos y segundos
17        int horas = segundosTotales / 3600;
18        int minutos = (segundosTotales % 3600) / 60;
19        int segundos = segundosTotales % 60;
20
21        // Mostrar los resultados
22        System.out.println("Horas: " + horas);
23        System.out.println("Minutos: " + minutos);
24        System.out.println("Segundos: " + segundos);
25
26        // Cerrar el objeto Scanner para evitar fugas de recursos
27        scanner.close();
28    }
29
30 }
```

```
Output - Ejercicio6 (run) x
run:
Ingrese la cantidad de segundos: 53247
Horas: 14
Minutos: 47
Segundos: 27
BUILD SUCCESSFUL (total time: 10 seconds)
```

## Ejercicio 1.17

1.17. Solicita al usuario tres distancias:

- La primera, medida en milímetros.
- La segunda, medida en centímetros.
- La última, medida en metros.

Diseña un programa que muestre la suma de las tres longitudes introducidas (medida en centímetros).

```
Source History 
7 public class Ejercicio7 {
8
9     public static void main(String[] args) {
10         // Crear un objeto Scanner para leer la entrada del usuario
11         Scanner scanner = new Scanner(System.in);
12
13         // Solicitar al usuario que ingrese las distancias en diferentes unidades
14         System.out.print("Ingrese la distancia en milímetros: ");
15         double milimetros = scanner.nextDouble();
16         System.out.print("Ingrese la distancia en centímetros: ");
17         double centimetros = scanner.nextDouble();
18         System.out.print("Ingrese la distancia en metros: ");
19         double metros = scanner.nextDouble();
20
21         // Calcular la suma en centímetros
22         double sumaCentimetros = convertirAMilimetros(milimetros) +
23             convertirACentimetros(centimetros) + convertirAMetros(metros);
24
25         // Mostrar la suma en centímetros
26         System.out.println("La suma de las distancias en centímetros es: "
27             + sumaCentimetros + " cm");
28
29         scanner.close();
30     }
31
32     // Método para convertir milímetros a centímetros
33     private static double convertirAMilimetros(double milimetros) {
34         return milimetros / 10;
35     }
36
37     // Método para convertir centímetros a centímetros (no es necesario convertir)
38     private static double convertirACentimetros(double centimetros) {
39         return centimetros;
40     }
41
42     // Método para convertir metros a centímetros
43     private static double convertirAMetros(double metros) {
44         return metros * 100;
45     }
46 }
```

### Output - Ejercicio7 (run) X

```
run:
Ingrese la distancia en milímetros: 950
Ingrese la distancia en centímetros: 30
Ingrese la distancia en metros: 1
La suma de las distancias en centímetros es: 225.0 cm
BUILD SUCCESSFUL (total time: 19 seconds)
```



## Ejercicio1.18

1.18. Un biólogo está realizando un estudio de distintas especies de invertebrados y necesita una aplicación que le ayude a contabilizar el número de patas que tienen en total todos los animales capturados durante una jornada de trabajo. Para ello, te ha solicitado que escribas una aplicación a la que hay que proporcionar:

- El número de hormigas capturadas (6 patas).
  - El número de arañas capturadas (8 patas).
  - El número de cochinillas capturadas (14 patas).
- La aplicación debe mostrar el número total de patas.

```
Source History 
1
2 package ejercicio8;
3
4 import java.util.Scanner;
5
6 public class Ejercicio8 {
7
8     public static void main(String[] args) {
9         Scanner scanner = new Scanner(System.in);
10
11         // Solicitar al usuario nº de hormigas
12         System.out.print("Ingrese nº de hormigas: ");
13         int hormigas = scanner.nextInt();
14         System.out.print("Ingrese nº de arañas: ");
15         int arañas = scanner.nextInt();
16         System.out.print("Ingrese nº de cochinillas: ");
17         int cochinillas = scanner.nextInt();
18         scanner.close();
19         // Calcular la suma de las patas
20         int sumaPatas = (hormigas * 6) + (arañas * 8) + (cochinillas * 14);
21
22         // Mostrar la suma en centímetros
23         System.out.println("La suma de las patas de insectos es: " + sumaPatas );
24     }
25 }
26
```

### Output - Ejercicio8 (run) ×

```
run:
Ingrese nº de hormigas: 5
Ingrese nº de arañas: 3
Ingrese nº de cochinillas: 2
La suma de las patas de insectos es: 82
BUILD SUCCESSFUL (total time: 15 seconds)
```



## Ejercicio1.19

- 1.19. Una empresa que gestiona un parque acuático te solicita una aplicación que les ayude a calcular el importe que hay que cobrar en la taquilla por la compra de una serie de entradas (cuyo número será introducido por el usuario). Existen dos tipos de entrada: infantiles, que cuestan 15,50 €; y de adultos, que cuestan 20 €.
- En el caso de que el importe total sea igual o superior a 100 €, se aplicará automáticamente un bono descuento del 5 %.

```
1
2 package ejercicio9;
3
4 import java.util.Scanner;
5
6
7 public class Ejercicio9 {
8
9
10     public static void main(String[] args) {
11         Scanner scanner = new Scanner(System.in);
12
13         // Solicitar al usuario nº de entradas
14         System.out.print("Ingrese nº de entradas infantiles: ");
15         int infantiles = scanner.nextInt();
16         System.out.print("Ingrese nº de entradas adultos: ");
17         int adultos = scanner.nextInt();
18         //calculo el precio sin descuento
19         double precio = (double)(infantiles * 15.50) + (adultos * 20);
20         //calculo el descuento si aplica
21         precio = (precio >= 100) ? (precio * 0.95) : precio;
22         System.out.println("El importe total es: " + precio);
23     }
24 }
25
26
```

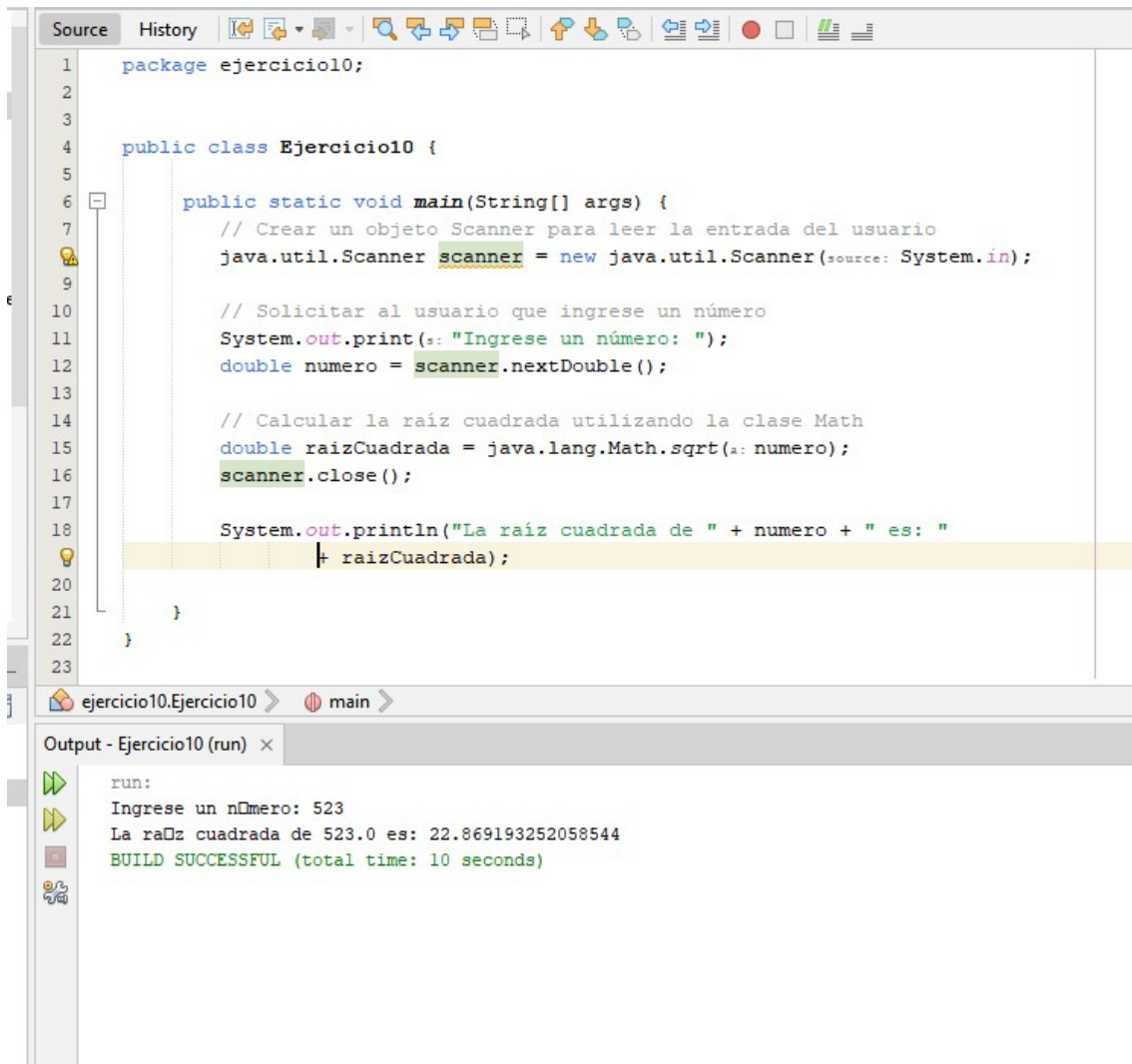
ejercicio9.Ejercicio9 > main >

Output - Ejercicio9 (run) x

run:  
Ingrese nº de entradas infantiles: 6  
Ingrese nº de entradas adultos: 1  
El importe total es: 107.35  
BUILD SUCCESSFUL (total time: 4 seconds)

## Ejercicio1.20

1.20. Solicita al usuario un número real y calcula su raíz cuadrada. Implementa el programa utilizando el nombre cualificado de las clases, en lugar de utilizar ninguna importación.



```
1 package ejercicio10;  
2  
3  
4 public class Ejercicio10 {  
5  
6     public static void main(String[] args) {  
7         // Crear un objeto Scanner para leer la entrada del usuario  
8         java.util.Scanner scanner = new java.util.Scanner(System.in);  
9  
10        // Solicitar al usuario que ingrese un número  
11        System.out.print("Ingrese un número: ");  
12        double numero = scanner.nextDouble();  
13  
14        // Calcular la raíz cuadrada utilizando la clase Math  
15        double raizCuadrada = java.lang.Math.sqrt(numero);  
16        scanner.close();  
17  
18        System.out.println("La raíz cuadrada de " + numero + " es: "  
19        + raizCuadrada);  
20  
21    }  
22 }  
23
```

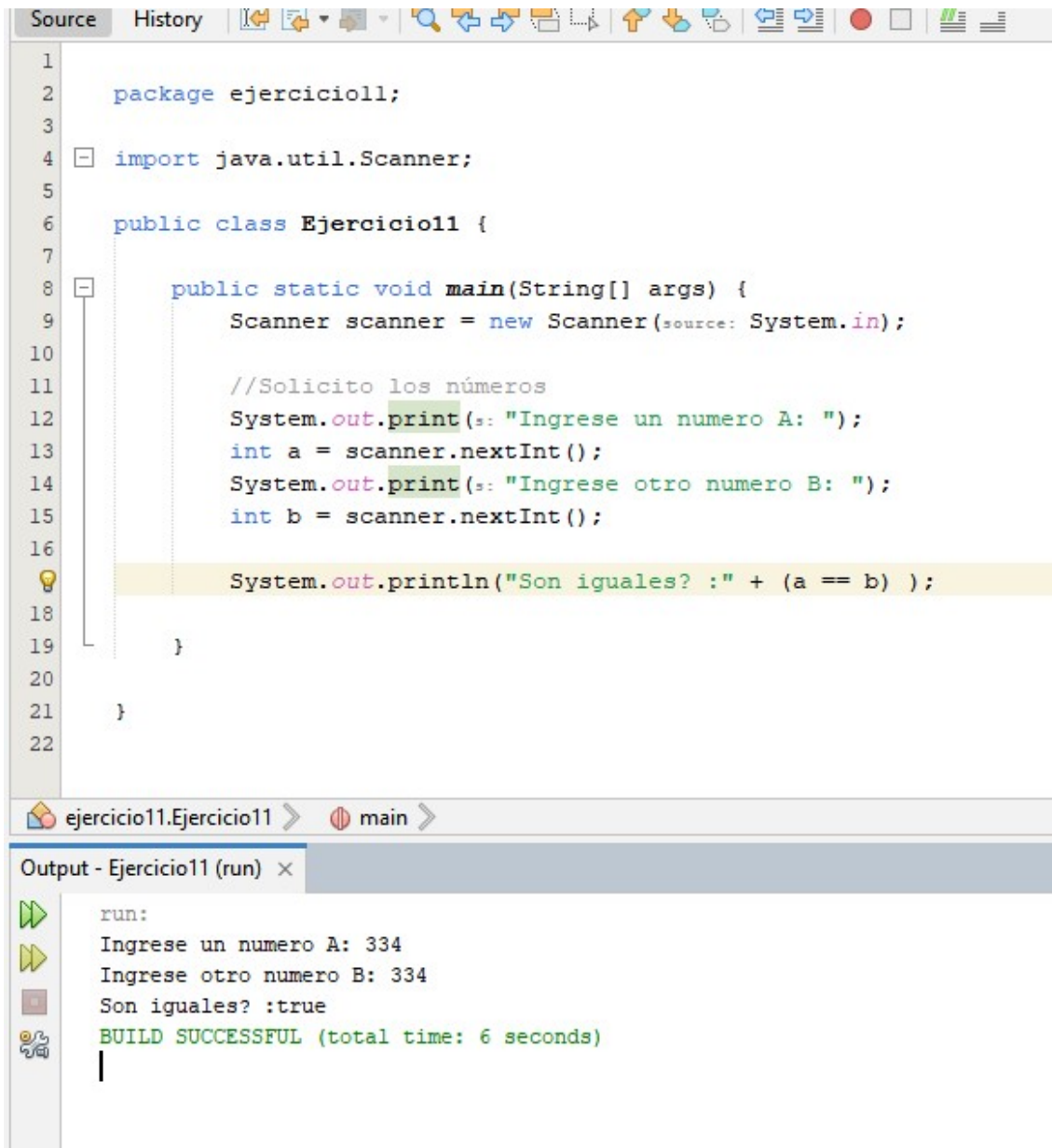
ejercicio10.Ejercicio10 > main >

Output - Ejercicio10 (run) x

```
run:  
Ingrese un número: 523  
La raíz cuadrada de 523.0 es: 22.869193252058544  
BUILD SUCCESSFUL (total time: 10 seconds)
```

## Ejercicio1.21

1.21. Pide dos números al usuario: *a* y *b*. Deberá mostrarse `true` si ambos números son iguales y `false` en caso contrario.



The screenshot shows an IDE with a Java source file named `Ejercicio11.java`. The code is as follows:

```
1 package ejerciciol1;
2
3
4 import java.util.Scanner;
5
6 public class Ejercicio11 {
7
8     public static void main(String[] args) {
9         Scanner scanner = new Scanner(System.in);
10
11         //Solicito los números
12         System.out.print("Ingrese un numero A: ");
13         int a = scanner.nextInt();
14         System.out.print("Ingrese otro numero B: ");
15         int b = scanner.nextInt();
16
17         System.out.println("Son iguales? :" + (a == b) );
18     }
19 }
20
21 }
22 }
```

The IDE's output window shows the following execution results:

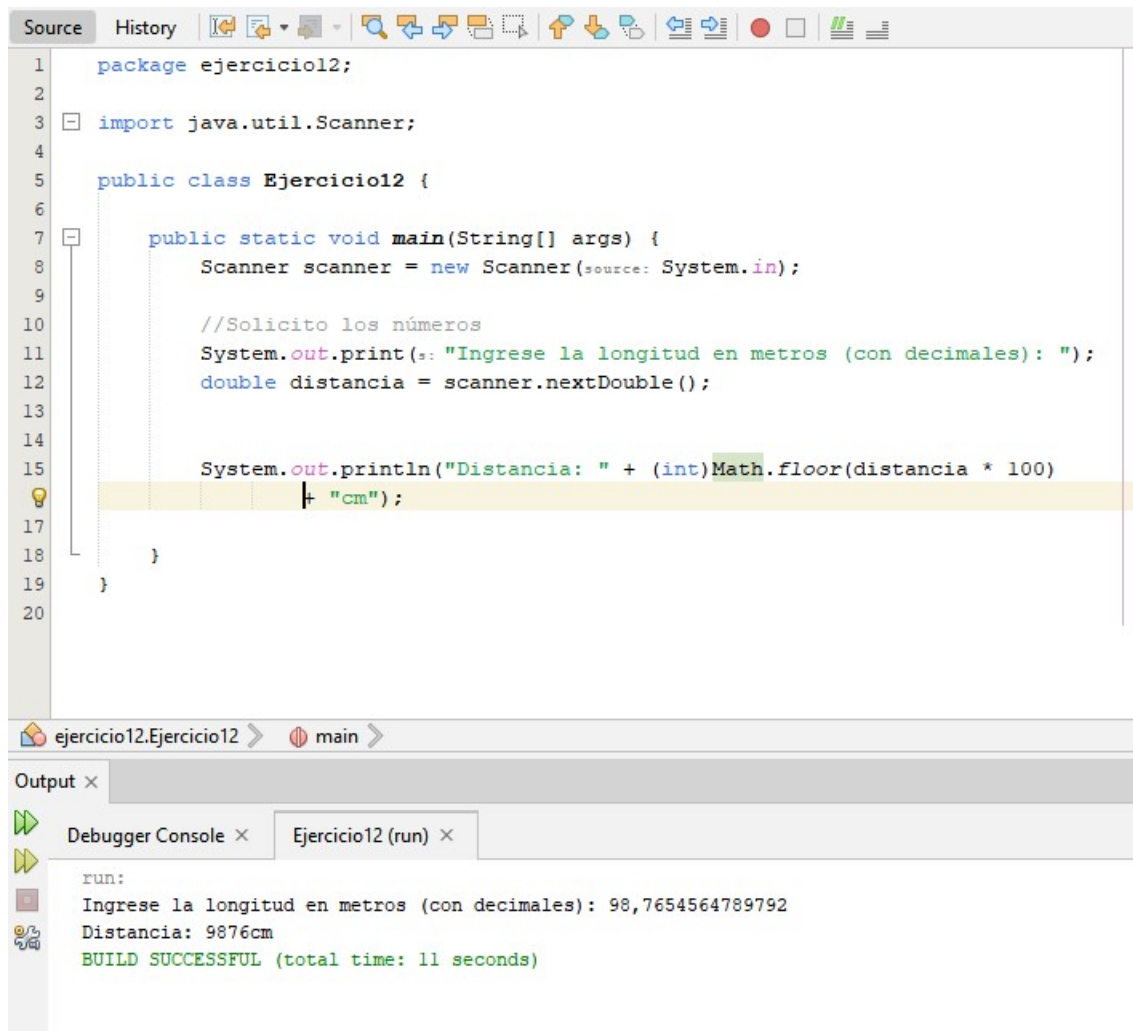
```
run:
Ingrese un numero A: 334
Ingrese otro numero B: 334
Son iguales? :true
BUILD SUCCESSFUL (total time: 6 seconds)
```



## Ejercicio1.22

**1.22.** La FILA (Federación Internacional de Lanzamiento de Algoritmo) realiza una competición donde cada participante escribe un algoritmo en un papel y lo lanza, ganando quien consiga lanzarlo más lejos. La peculiaridad del concurso es que la longitud del lanzamiento se mide en metros (con tantos decimales como se desee), pero para el ranking solo se tiene en cuenta la longitud en centímetros (sin decimales). Por ejemplo, para un lanzamiento de 12,3456 m (que son 1234,56 cm) solo se contabilizarán 1234 cm.

Realiza un programa que solicite la longitud (en metros) de un lanzamiento y muestre la parte entera correspondiente en centímetros.



```
1 package ejercicio12;  
2  
3 import java.util.Scanner;  
4  
5 public class Ejercicio12 {  
6  
7     public static void main(String[] args) {  
8         Scanner scanner = new Scanner(System.in);  
9  
10        //Solicito los números  
11        System.out.print("Ingrese la longitud en metros (con decimales): ");  
12        double distancia = scanner.nextDouble();  
13  
14  
15        System.out.println("Distancia: " + (int) Math.floor(distancia * 100)  
16                           + "cm");  
17    }  
18 }  
19  
20
```

ejercicio12.Ejercicio12 > main >

Output x

Debugger Console x Ejercicio12 (run) x

run:  
Ingrese la longitud en metros (con decimales): 98,7654564789792  
Distancia: 9876cm  
BUILD SUCCESSFUL (total time: 11 seconds)