

CIFP César Manrique.

Programación 1º de Desarrollo de Aplicaciones Web

Profesor: José David Díaz Díaz

Actividades



Esta obra está licenciada bajo la Licencia Creative Commons Atribución 4.0 Internacional. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by/4.0/> o envíe una carta a Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

Índice

Actividades.....	1
Desarrollo.....	2
Actividades de comprobación.....	2
1.1 ¿Cuál de los siguientes identificadores no puede emplearse en una variable?.....	2
1.2 De todos los tipos primitivos disponibles en Java, selecciona cuál o cuáles son los que tienen un mayor tamaño y, por lo tanto, pueden albergar un mayor número de valores:.....	2
1.3 ¿Mediante que símbolo es posible añadir un comentario en nuestro código?.....	2
1.4 ¿Qué paquete se importa automáticamente en cualquier programa sin necesidad de tener que utilizar la sentencia import?.....	2
1.5 ¿Cuál de las siguientes instrucciones nos permiten mostrar información por consola?.....	2
1.6 ¿Qué instrucción equivale a i++?.....	2
1.7 Si evalúas la siguiente expresión $2 < 1 \parallel 2 != 1$, el resultado de dicha expresión es:.....	3
1.8 ¿Qué valor toma la variable a, tras la ejecución de la instrucción:.....	3
int a = 1 < 2 ? 3 : 4.....	3
1.9 Selecciona la expresión cuya evaluación resulta 3:.....	3
1.10 En las siguientes conversiones de tipo, ¿cuál de ellas produce un error?.....	3
Actividades de Aplicación.....	4
Actividad 1.11.....	4
Actividad 1.12.....	5
Actividad 1.13.....	6
Actividad 1.14.....	7
Actividad 1.15.....	8
Actividad 1.16.....	9

Actividad 1.17.....	10
Actividad 1.18.....	11
Actividad 1.19.....	12
Actividad 1.20.....	13
Actividad 1.21.....	14
Actividad 1.22.....	15

Actividades

Actividades de la Unidad 1: Conceptos Básicos.

En este documento se detallan las actividades a realizar. Se entregará al profesor en la plataforma digital dos ficheros. Un primer fichero pdf con todas las actividades a realizar, el nombre del fichero será “unidad1 + nombre del alumno.pdf”. Añadir en el fichero pdf por cada actividad de programación dos capturas de pantalla, una del código y otra de su ejecución. También en el fichero pdf copiar todas las preguntas y las respuestas correctas de las actividades de comprobación. Además, entregar un segundo fichero comprimido con todos los códigos fuentes de cada actividad de programación realizada.

Todas las actividades resueltas se deberán de analizar y no se entregarán.

A continuación, detallamos las actividades a realizar:

- **Actividades propuestas.** Esta unidad no tiene dichas actividades.
- **Actividades de comprobación.** Realizarlas todas. Copiar las preguntas y las respuestas correctas.
- **Actividades de aplicación.** Realizar las siguientes **1.11, 1.12, 1.13, 1.14, 1.15, 1.16, 1.17, 1.18, 1.19, 1.20, 1.21 y 1.22.**
- **Actividades de ampliación.** No realizar ninguna.

[Volver al índice](#)

Desarrollo

Actividades de comprobación.

1.1 ¿Cuál de los siguientes identificadores no puede emplearse en una variable?

c) final

[Volver al índice](#)

1.2 De todos los tipos primitivos disponibles en Java, selecciona cuál o cuáles son los que tienen un mayor tamaño y, por lo tanto, pueden albergar un mayor número de valores:

b) long y double

[Volver al índice](#)

1.3 ¿Mediante que símbolo es posible añadir un comentario en nuestro código?

b) //

[Volver al índice](#)

1.4 ¿Qué paquete se importa automáticamente en cualquier programa sin necesidad de tener que utilizar la sentencia import?

d) java.lang

[Volver al índice](#)

1.5 ¿Cuál de las siguientes instrucciones nos permiten mostrar información por consola?

c) System.out.println()

[Volver al índice](#)

1.6 ¿Qué instrucción equivale a i++?

d) Cualquiera de los anteriores

[Volver al índice](#)

1.7 Si evalúas la siguiente expresión `2 < 1 || 2 != 1`, el resultado de dicha expresión es:

c) true

[Volver al índice](#)

1.8 ¿Qué valor toma la variable `a`, tras la ejecución de la instrucción:

```
int a = 1 < 2 ? 3 : 4
```

c) 3

[Volver al índice](#)

1.9 Selecciona la expresión cuya evaluación resulta 3:

c) $(3 + 2 * 6) / 5$

[Volver al índice](#)

1.10 En las siguientes conversiones de tipo, ¿cuál de ellas produce un error?

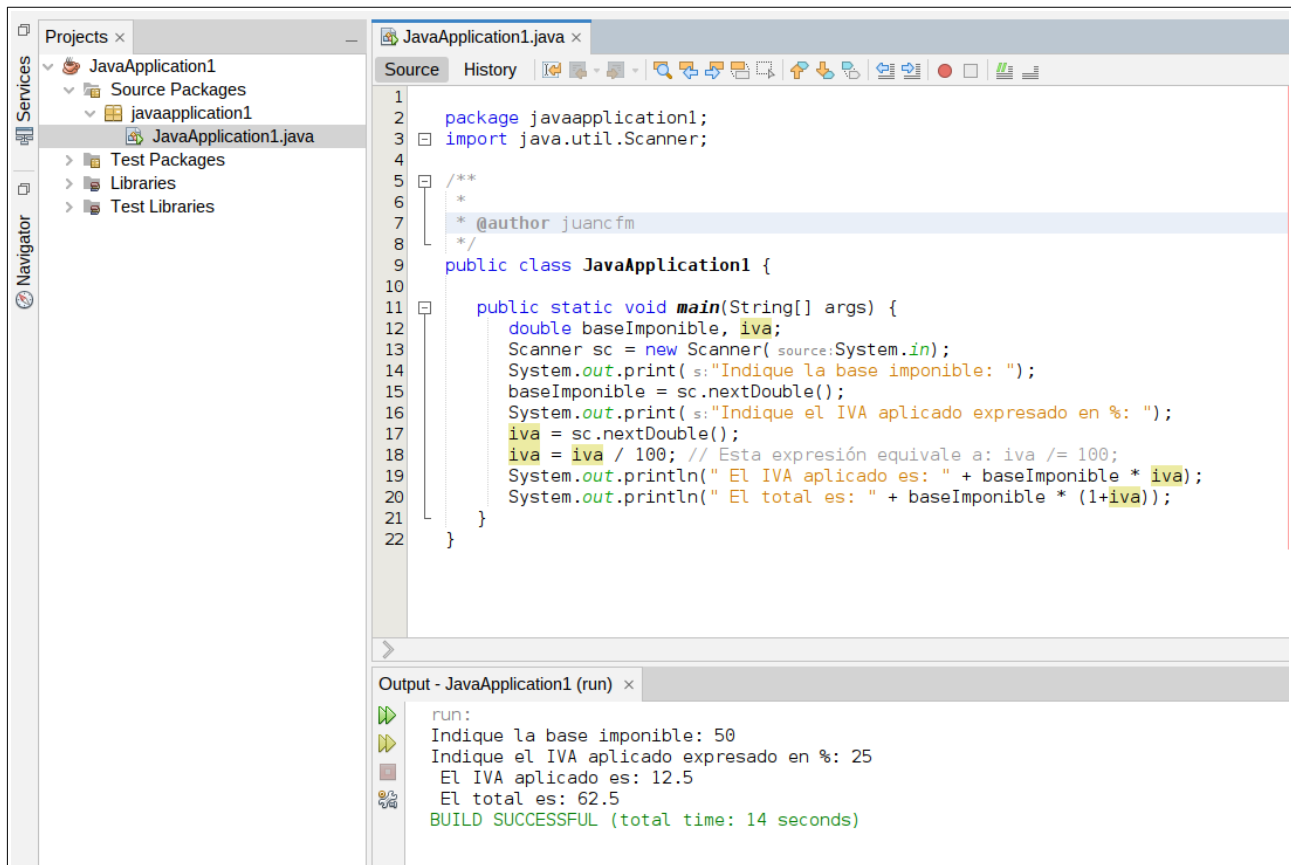
b) `int a = 12.3;`

[Volver al índice](#)

Actividades de Aplicación.

Actividad 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.



```
1 package javaapplication1;
2 import java.util.Scanner;
3
4 /**
5  *
6  * @author juancfm
7  */
8
9 public class JavaApplication1 {
10
11     public static void main(String[] args) {
12         double baseImponible, iva;
13         Scanner sc = new Scanner(System.in);
14         System.out.print("Indique la base imponible: ");
15         baseImponible = sc.nextDouble();
16         System.out.print("Indique el IVA aplicado expresado en %: ");
17         iva = sc.nextDouble();
18         iva = iva / 100; // Esta expresión equivale a: iva /= 100;
19         System.out.println(" El IVA aplicado es: " + baseImponible * iva);
20         System.out.println(" El total es: " + baseImponible * (1+iva));
21     }
22 }
```

Output - JavaApplication1 (run) x

```
run:
Indique la base imponible: 50
Indique el IVA aplicado expresado en %: 25
El IVA aplicado es: 12.5
El total es: 62.5
BUILD SUCCESSFUL (total time: 14 seconds)
```

[Volver al índice](#)

Actividades

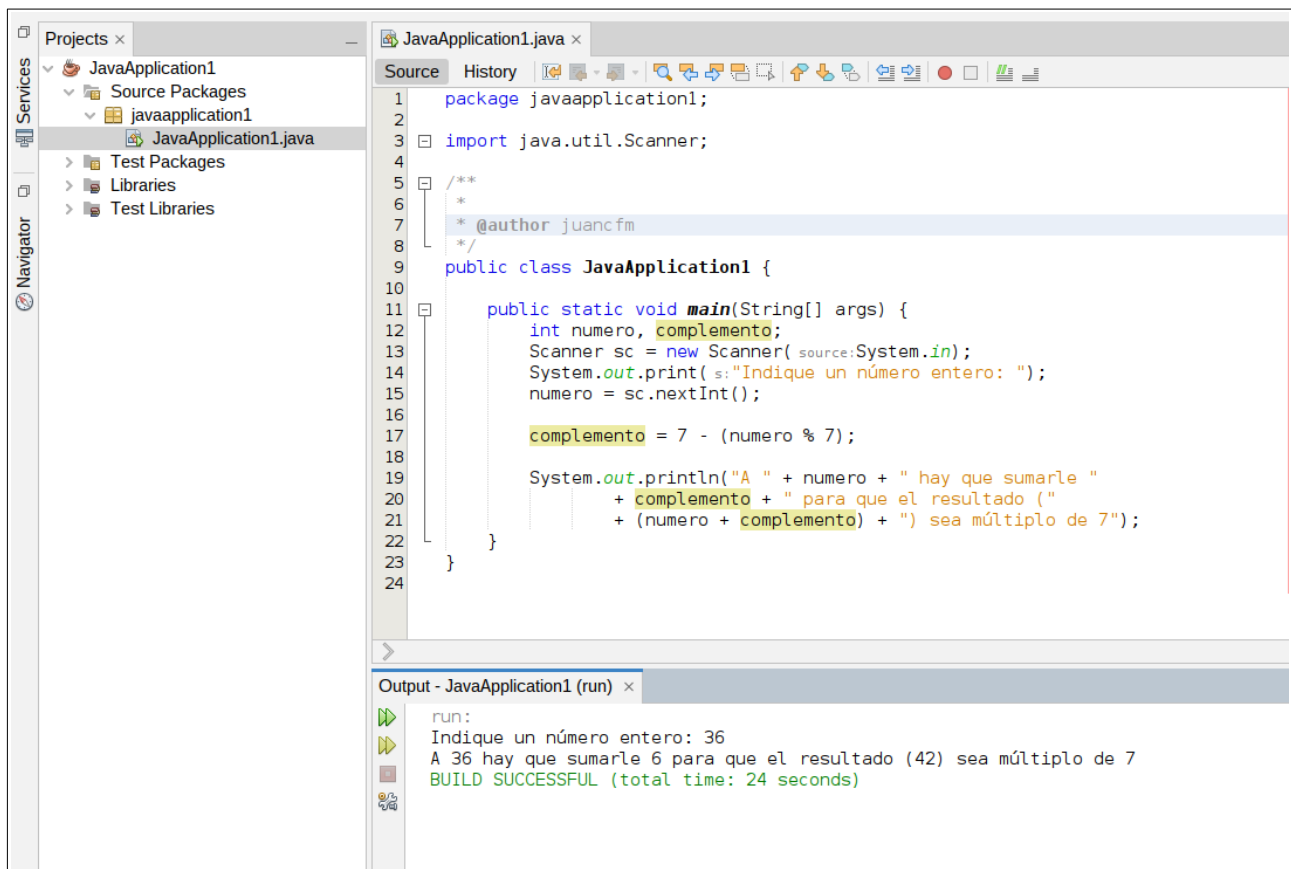
Actividad 1.12.

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.



The screenshot shows an IDE with a project named 'JavaApplication1'. The 'Source' tab is active, displaying the code for 'JavaApplication1.java'. The code uses a Scanner to take input and calculates the complement to make the sum a multiple of 7. The 'Output' window shows the execution result for the input 36, indicating a complement of 6 is needed to reach 42, which is a multiple of 7. The build was successful.

```
1 package javaapplication1;
2
3 import java.util.Scanner;
4
5 /**
6  * @author juancfm
7  */
8
9 public class JavaApplication1 {
10
11     public static void main(String[] args) {
12         int numero, complemento;
13         Scanner sc = new Scanner(System.in);
14         System.out.print("Indique un número entero: ");
15         numero = sc.nextInt();
16
17         complemento = 7 - (numero % 7);
18
19         System.out.println("A " + numero + " hay que sumarle "
20             + complemento + " para que el resultado ("
21             + (numero + complemento) + ") sea múltiplo de 7");
22     }
23 }
24
```

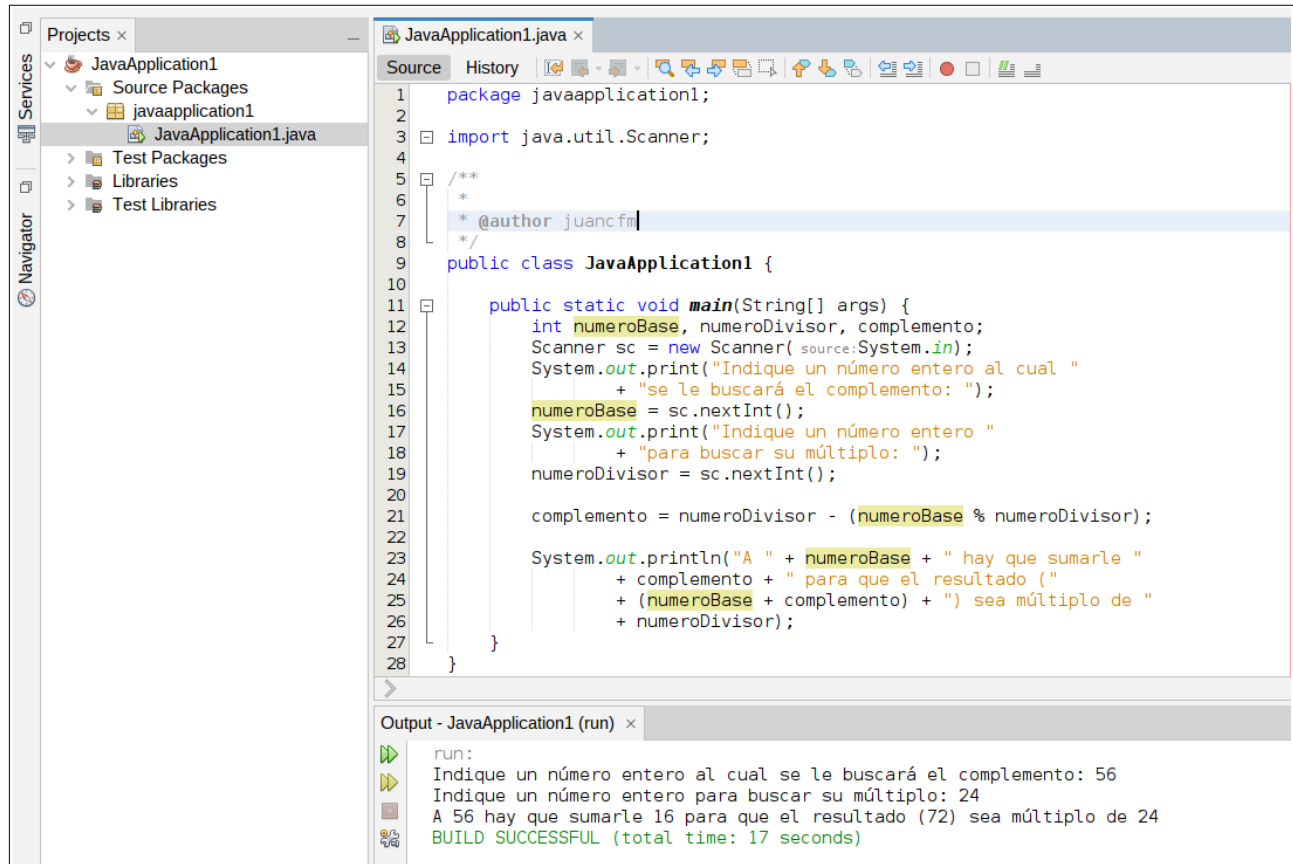
Output - JavaApplication1 (run) x

```
run:
Indique un número entero: 36
A 36 hay que sumarle 6 para que el resultado (42) sea múltiplo de 7
BUILD SUCCESSFUL (total time: 24 seconds)
```

[Volver al índice](#)

Actividad 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 .



```
1 package javaapplication1;
2
3 import java.util.Scanner;
4
5 /**
6  *
7  * @author juancfm
8  */
9 public class JavaApplication1 {
10
11     public static void main(String[] args) {
12         int numeroBase, numeroDivisor, complemento;
13         Scanner sc = new Scanner(System.in);
14         System.out.print("Indique un número entero al cual "
15             + "se le buscará el complemento: ");
16         numeroBase = sc.nextInt();
17         System.out.print("Indique un número entero "
18             + "para buscar su múltiplo: ");
19         numeroDivisor = sc.nextInt();
20
21         complemento = numeroDivisor - (numeroBase % numeroDivisor);
22
23         System.out.println("A " + numeroBase + " hay que sumarle "
24             + complemento + " para que el resultado ("
25             + (numeroBase + complemento) + ") sea múltiplo de "
26             + numeroDivisor);
27     }
28 }
```

Output - JavaApplication1 (run) ×

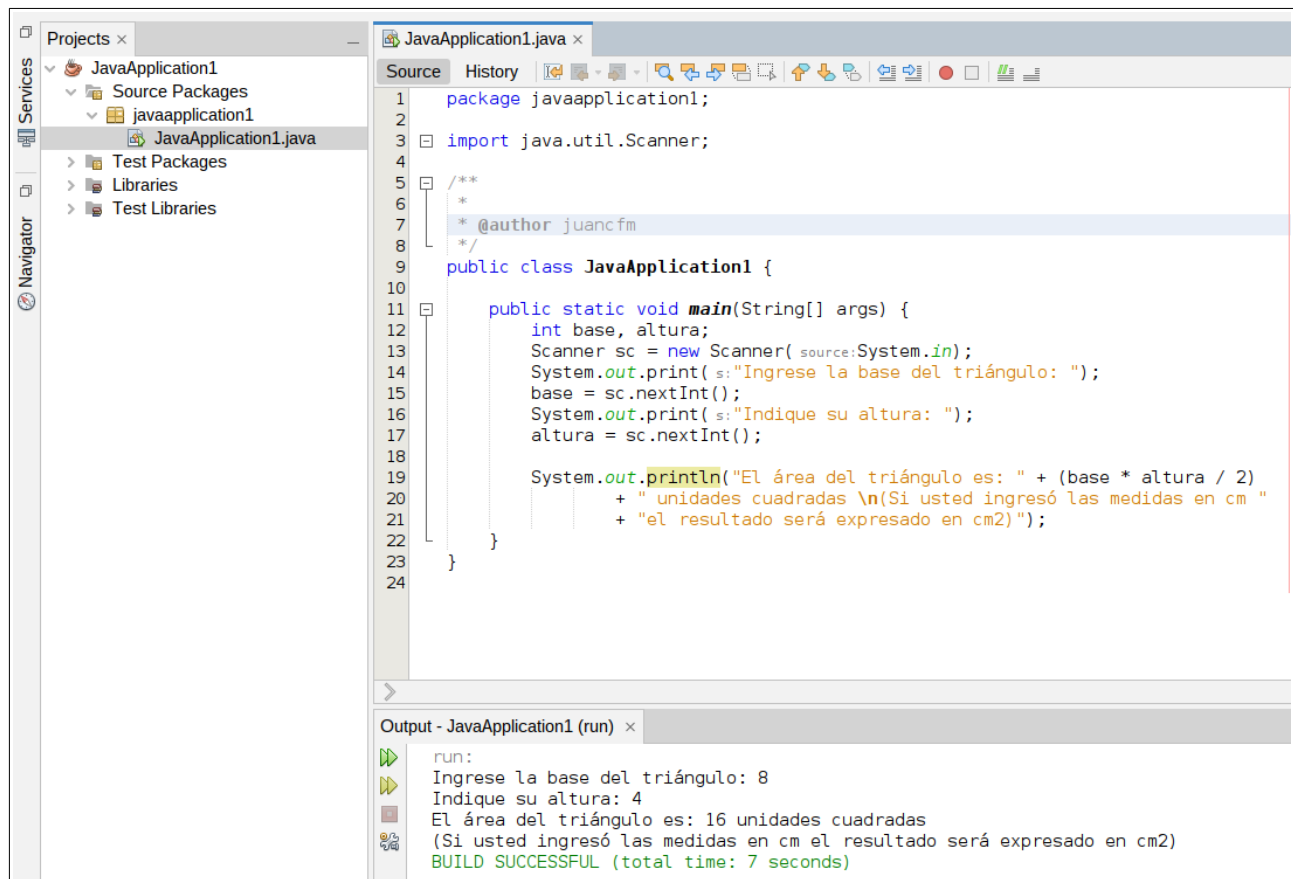
```
run:
Indique un número entero al cual se le buscará el complemento: 56
Indique un número entero para buscar su múltiplo: 24
A 56 hay que sumarle 16 para que el resultado (72) sea múltiplo de 24
BUILD SUCCESSFUL (total time: 17 seconds)
```

[Volver al índice](#)

Actividad 1.14.

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

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



The screenshot shows an IDE with a project named 'JavaApplication1'. The 'Source' editor displays the following Java code:

```
1 package javaapplication1;
2
3 import java.util.Scanner;
4
5 /**
6  *
7  * @author juancfm
8  */
9 public class JavaApplication1 {
10
11     public static void main(String[] args) {
12         int base, altura;
13         Scanner sc = new Scanner(System.in);
14         System.out.print("Ingrese la base del triángulo: ");
15         base = sc.nextInt();
16         System.out.print("Indique su altura: ");
17         altura = sc.nextInt();
18
19         System.out.println("El área del triángulo es: " + (base * altura / 2)
20             + " unidades cuadradas \n(Si usted ingresó las medidas en cm "
21             + "el resultado será expresado en cm2)");
22     }
23 }
24
```

The 'Output' window shows the execution results:

```
run:
Ingrese la base del triángulo: 8
Indique su altura: 4
El área del triángulo es: 16 unidades cuadradas
(Si usted ingresó las medidas en cm el resultado será expresado en cm2)
BUILD SUCCESSFUL (total time: 7 seconds)
```

[Volver al índice](#)

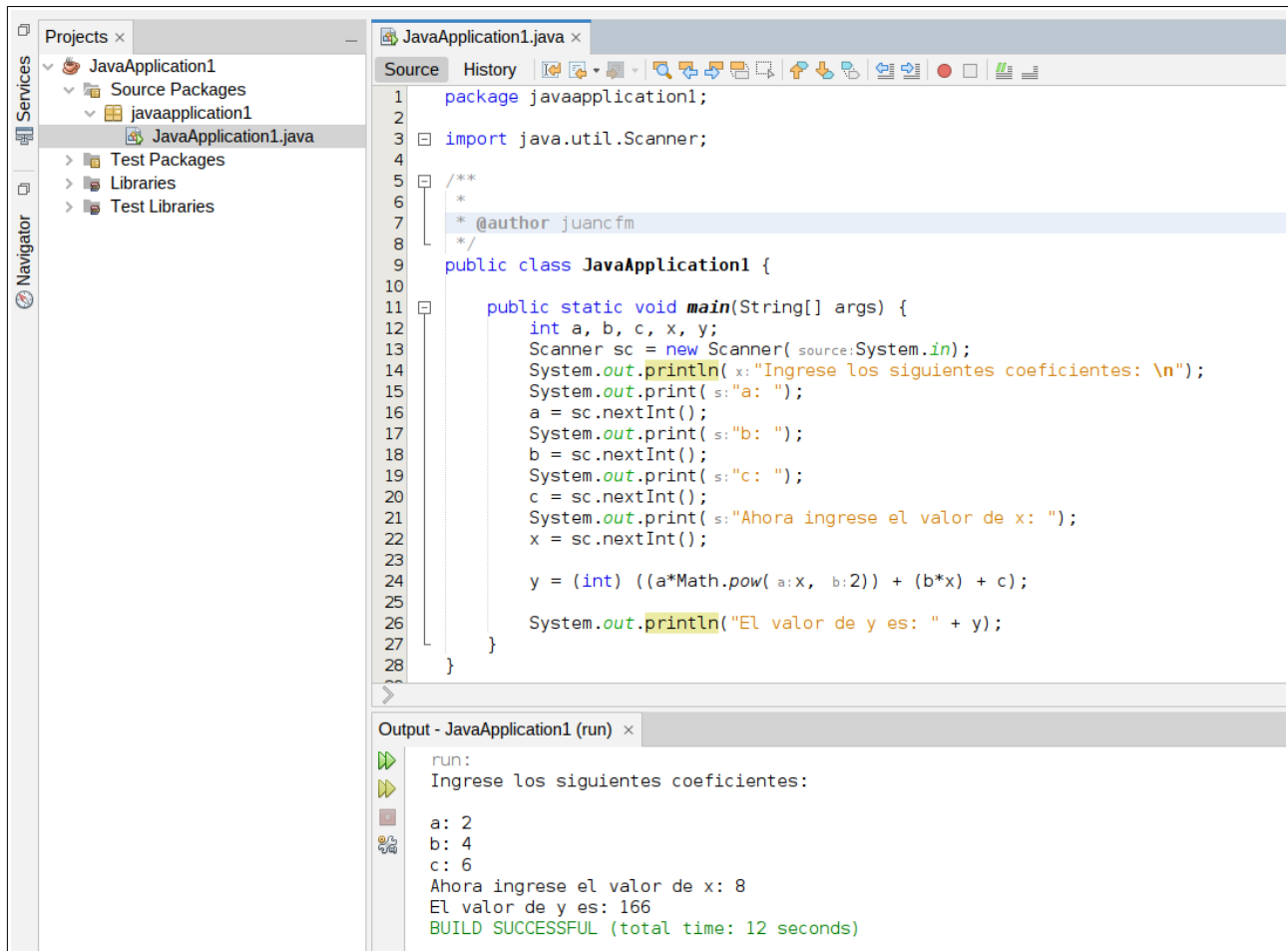
Actividades

Actividad 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 calcula el valor correspondiente de y.



The screenshot shows an IDE with a project named 'JavaApplication1'. The main editor displays the source code for 'JavaApplication1.java'. The code imports 'java.util.Scanner' and defines a 'main' method that prompts the user for coefficients 'a', 'b', and 'c', and a value 'x'. It then calculates 'y' using the formula $y = ax^2 + bx + c$ and prints the result. The output window shows the execution results for the input values a=2, b=4, c=6, and x=8, resulting in y=166.

```
1 package javaapplication1;
2
3 import java.util.Scanner;
4
5 /**
6  *
7  * @author juancfm
8  */
9 public class JavaApplication1 {
10
11     public static void main(String[] args) {
12         int a, b, c, x, y;
13         Scanner sc = new Scanner(System.in);
14         System.out.println("Ingrese los siguientes coeficientes: \n");
15         System.out.print("a: ");
16         a = sc.nextInt();
17         System.out.print("b: ");
18         b = sc.nextInt();
19         System.out.print("c: ");
20         c = sc.nextInt();
21         System.out.print("Ahora ingrese el valor de x: ");
22         x = sc.nextInt();
23
24         y = (int) ((a*Math.pow(a,x, b:2)) + (b*x) + c);
25
26         System.out.println("El valor de y es: " + y);
27     }
28 }
```

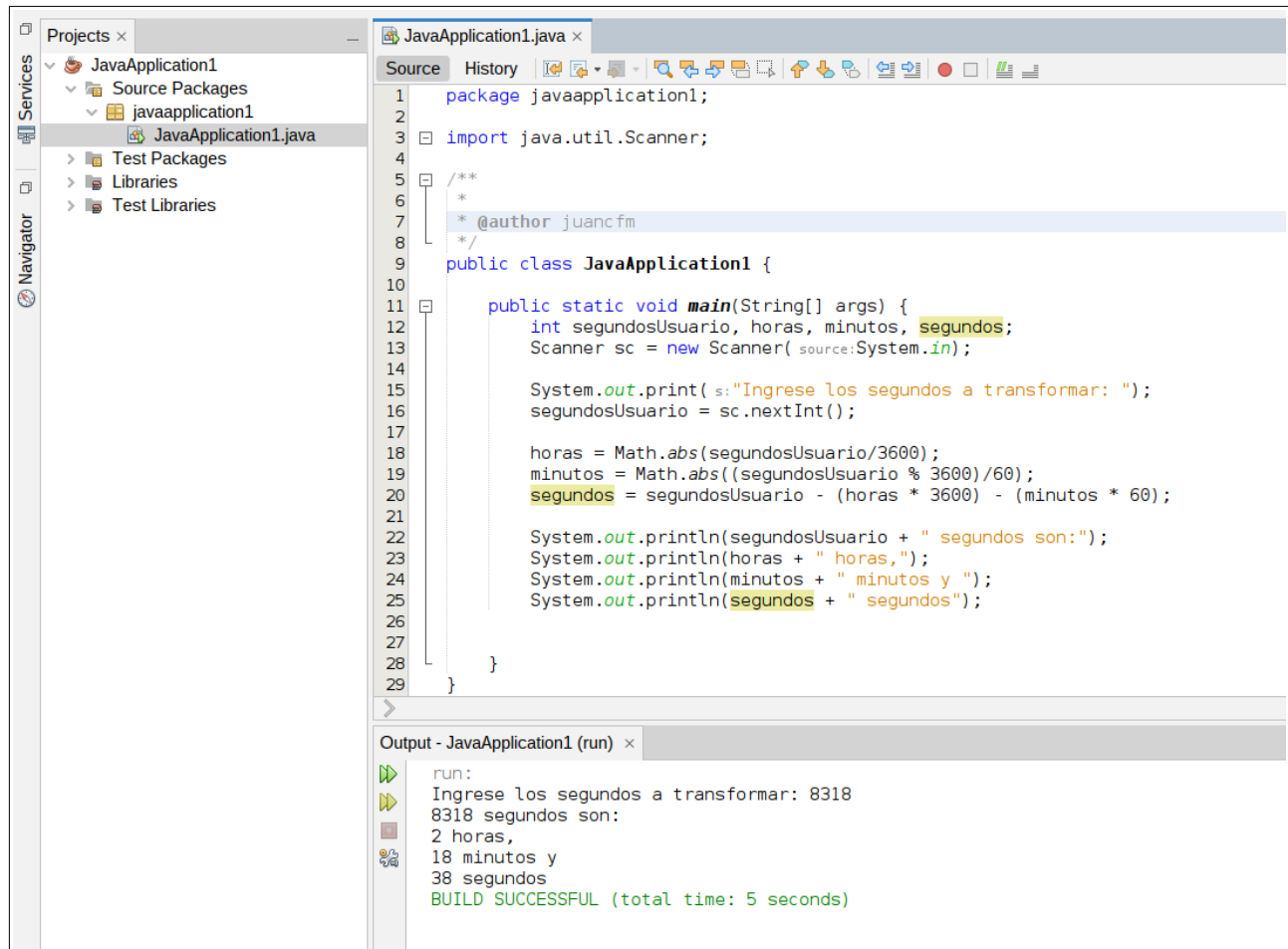
Output - JavaApplication1 (run) x

```
run:
Ingrese los siguientes coeficientes:
a: 2
b: 4
c: 6
Ahora ingrese el valor de x: 8
El valor de y es: 166
BUILD SUCCESSFUL (total time: 12 seconds)
```

[Volver al índice](#)

Actividad 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.



The screenshot shows an IDE with a project named 'JavaApplication1'. The source code for 'JavaApplication1.java' is displayed, featuring a package declaration, an import for 'Scanner', and a class with a 'main' method. The code prompts the user for seconds, calculates the equivalent hours, minutes, and seconds, and prints the results. The output window shows the execution with an input of 8318 seconds, resulting in 2 hours, 18 minutes, and 38 seconds.

```
1 package javaapplication1;
2
3 import java.util.Scanner;
4
5 /**
6  *
7  * @author juancfm
8  */
9 public class JavaApplication1 {
10
11     public static void main(String[] args) {
12         int segundosUsuario, horas, minutos, segundos;
13         Scanner sc = new Scanner(System.in);
14
15         System.out.print("Ingrese los segundos a transformar: ");
16         segundosUsuario = sc.nextInt();
17
18         horas = Math.abs(segundosUsuario/3600);
19         minutos = Math.abs((segundosUsuario % 3600)/60);
20         segundos = segundosUsuario - (horas * 3600) - (minutos * 60);
21
22         System.out.println(segundosUsuario + " segundos son:");
23         System.out.println(horas + " horas,");
24         System.out.println(minutos + " minutos y ");
25         System.out.println(segundos + " segundos");
26
27     }
28 }
29
```

Output - JavaApplication1 (run) x

```
run:
Ingrese los segundos a transformar: 8318
8318 segundos son:
2 horas,
18 minutos y
38 segundos
BUILD SUCCESSFUL (total time: 5 seconds)
```

[Volver al índice](#)

Actividades

Actividad 1.17.

Solicita al usuario tres distancias:

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

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

```
1 package javaapplication1;
2
3 import java.util.Scanner;
4
5 /**
6  *
7  * @author juancfm
8  */
9 public class JavaApplication1 {
10
11     public static void main(String[] args) {
12         double primeraMedida, segundaMedida, terceraMedida;
13         Scanner sc = new Scanner(System.in);
14
15         System.out.print("Ingrese la primera medida de longitud "
16             + "(expresada en milímetros): ");
17         primeraMedida = sc.nextDouble();
18
19         System.out.print("Ingrese la segunda medida de longitud "
20             + "(expresada en centímetros): ");
21         segundaMedida = sc.nextDouble();
22
23         System.out.print("Ingrese la tercera medida de longitud "
24             + "(expresada en metros): ");
25         terceraMedida = sc.nextDouble();
26
27         primeraMedida = primeraMedida / 10;
28         terceraMedida = terceraMedida * 100;
29
30         sc.close();
31
32         System.out.println("La suma de las tres longitudes introducidas es: "
33             + (primeraMedida + segundaMedida + terceraMedida)
34             + "centímetros");
35     }
36 }
37
38 javaapplication1.JavaApplication1 > main >
```

Output - JavaApplication1 (run) ×

```
run:
Ingrese la primera medida de longitud (expresada en milímetros): 45
Ingrese la segunda medida de longitud (expresada en centímetros): 123
Ingrese la tercera medida de longitud (expresada en metros): 3,4
La suma de las tres longitudes introducidas es: 467.5centímetros
BUILD SUCCESSFUL (total time: 17 seconds)
```

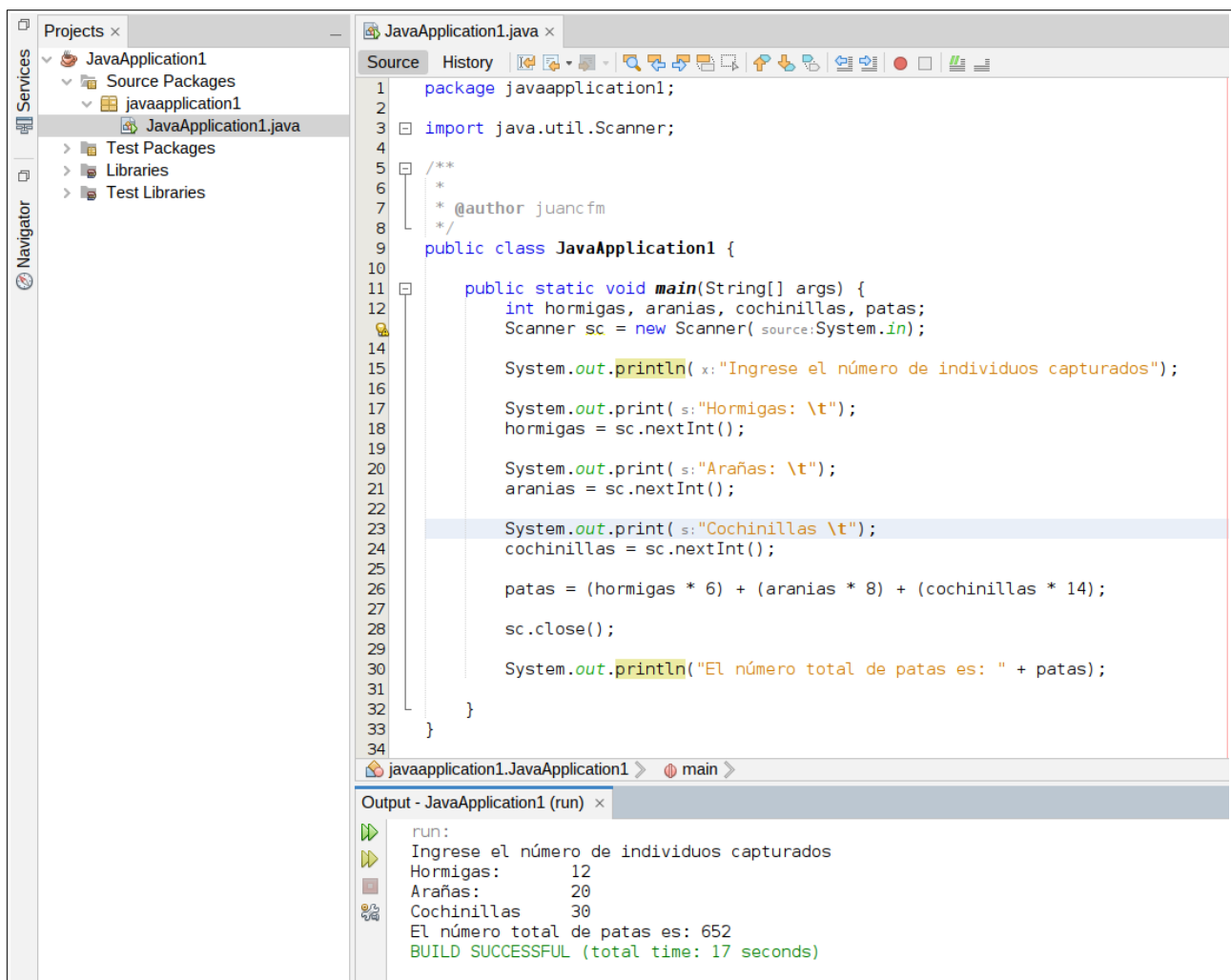
[Volver al índice](#)

Actividad 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.



The screenshot shows an IDE with a project named 'JavaApplication1'. The source file 'JavaApplication1.java' is open, displaying the following code:

```
1 package javaapplication1;
2
3 import java.util.Scanner;
4
5 /**
6  *
7  * @author juancfm
8  */
9 public class JavaApplication1 {
10
11     public static void main(String[] args) {
12         int hormigas, arañas, cochinillas, patas;
13         Scanner sc = new Scanner(System.in);
14
15         System.out.println("Ingrese el número de individuos capturados");
16
17         System.out.print("Hormigas: ");
18         hormigas = sc.nextInt();
19
20         System.out.print("Arañas: ");
21         arañas = sc.nextInt();
22
23         System.out.print("Cochinillas ");
24         cochinillas = sc.nextInt();
25
26         patas = (hormigas * 6) + (arañas * 8) + (cochinillas * 14);
27
28         sc.close();
29
30         System.out.println("El número total de patas es: " + patas);
31     }
32 }
33
34
```

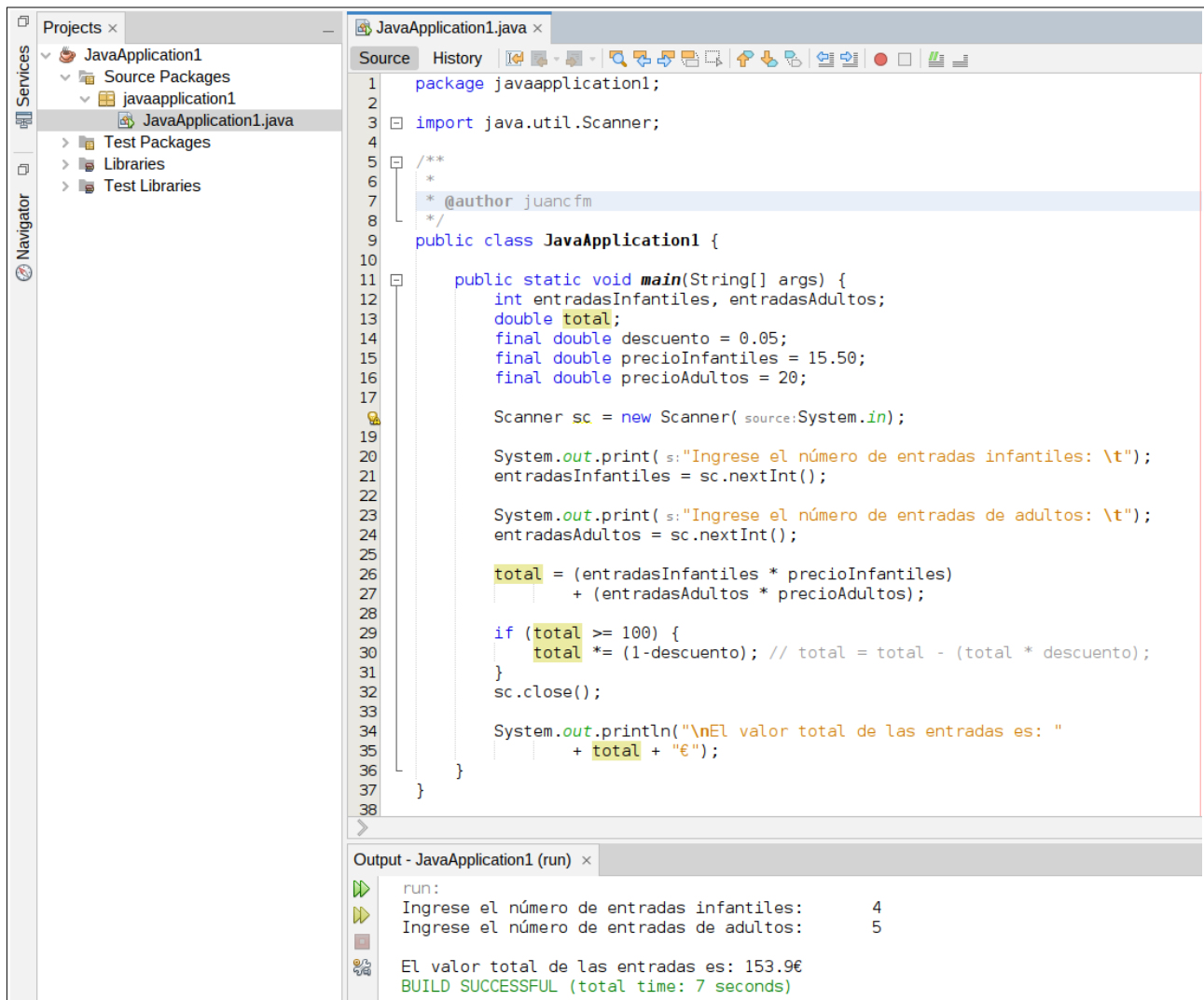
The output window shows the execution results:

```
run:
Ingrese el número de individuos capturados
Hormigas:      12
Arañas:        20
Cochinillas    30
El número total de patas es: 652
BUILD SUCCESSFUL (total time: 17 seconds)
```

[Volver al índice](#)

Actividad 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 %.



The screenshot displays an IDE with the following components:

- Projects:** A tree view on the left showing the project structure: `JavaApplication1` (containing `Source Packages` and `Test Packages`), `javaapplication1` (containing `JavaApplication1.java`), `Test Packages`, `Libraries`, and `Test Libraries`.
- Source:** The main editor window showing the code for `JavaApplication1.java`. The code is as follows:

```
1 package javaapplication1;
2
3 import java.util.Scanner;
4
5 /**
6  *
7  * @author juancfm
8  */
9 public class JavaApplication1 {
10
11     public static void main(String[] args) {
12         int entradasInfantiles, entradasAdultos;
13         double total;
14         final double descuento = 0.05;
15         final double precioInfantiles = 15.50;
16         final double precioAdultos = 20;
17
18         Scanner sc = new Scanner( source: System.in );
19
20         System.out.print( s: "Ingrese el número de entradas infantiles: \t" );
21         entradasInfantiles = sc.nextInt();
22
23         System.out.print( s: "Ingrese el número de entradas de adultos: \t" );
24         entradasAdultos = sc.nextInt();
25
26         total = (entradasInfantiles * precioInfantiles)
27             + (entradasAdultos * precioAdultos);
28
29         if (total >= 100) {
30             total *= (1 - descuento); // total = total - (total * descuento);
31         }
32         sc.close();
33
34         System.out.println( "\nEl valor total de las entradas es: "
35             + total + "€" );
36     }
37 }
38
```
- Output - JavaApplication1 (run):** A console window at the bottom showing the execution results:

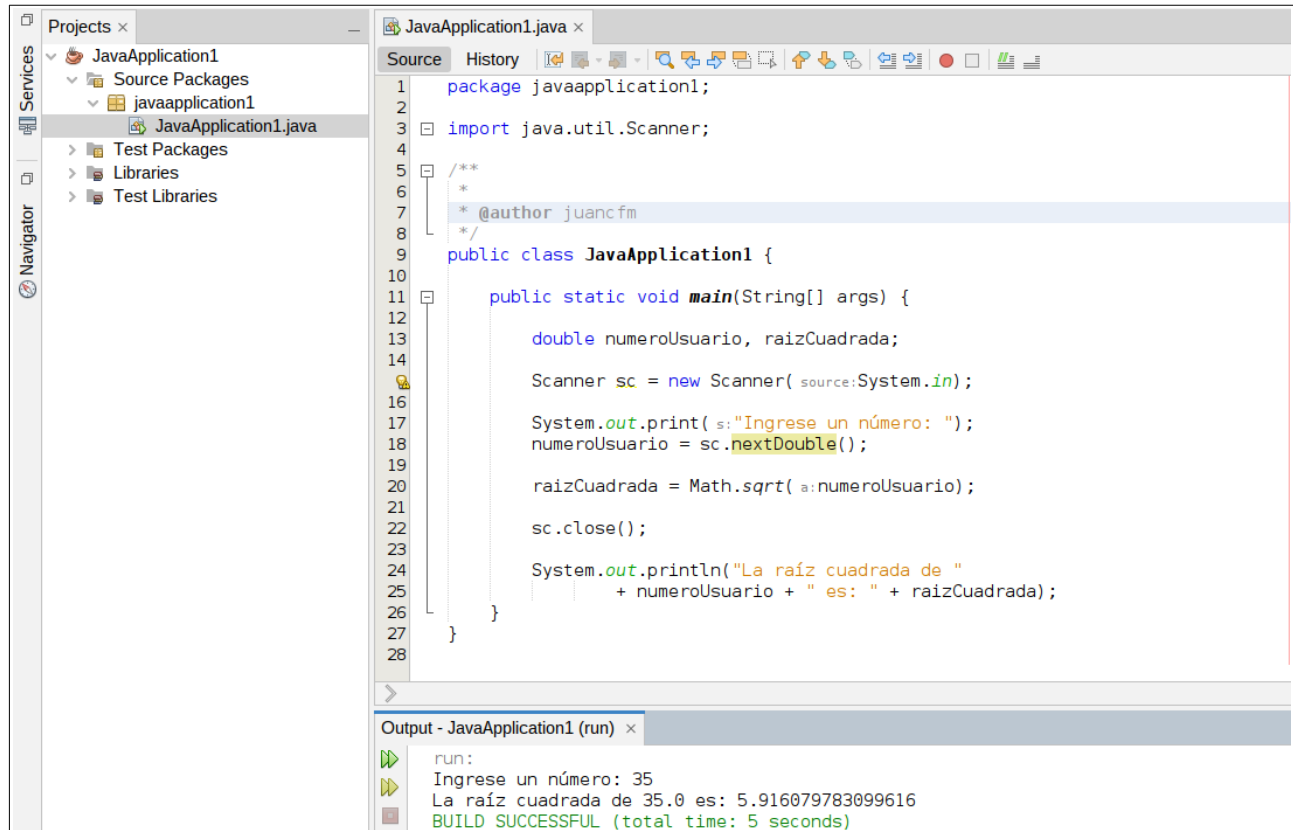
```
run:
Ingrese el número de entradas infantiles: 4
Ingrese el número de entradas de adultos: 5

El valor total de las entradas es: 153.9€
BUILD SUCCESSFUL (total time: 7 seconds)
```

[Volver al índice](#)

Actividad 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 javaapplication1;
2
3 import java.util.Scanner;
4
5 /**
6  *
7  * @author juancfm
8  */
9 public class JavaApplication1 {
10
11     public static void main(String[] args) {
12
13         double numeroUsuario, raizCuadrada;
14
15         Scanner sc = new Scanner( source: System.in);
16
17         System.out.print( s: "Ingrese un número: ");
18         numeroUsuario = sc.nextDouble();
19
20         raizCuadrada = Math.sqrt( a: numeroUsuario);
21
22         sc.close();
23
24         System.out.println("La raíz cuadrada de "
25             + numeroUsuario + " es: " + raizCuadrada);
26     }
27 }
28
```

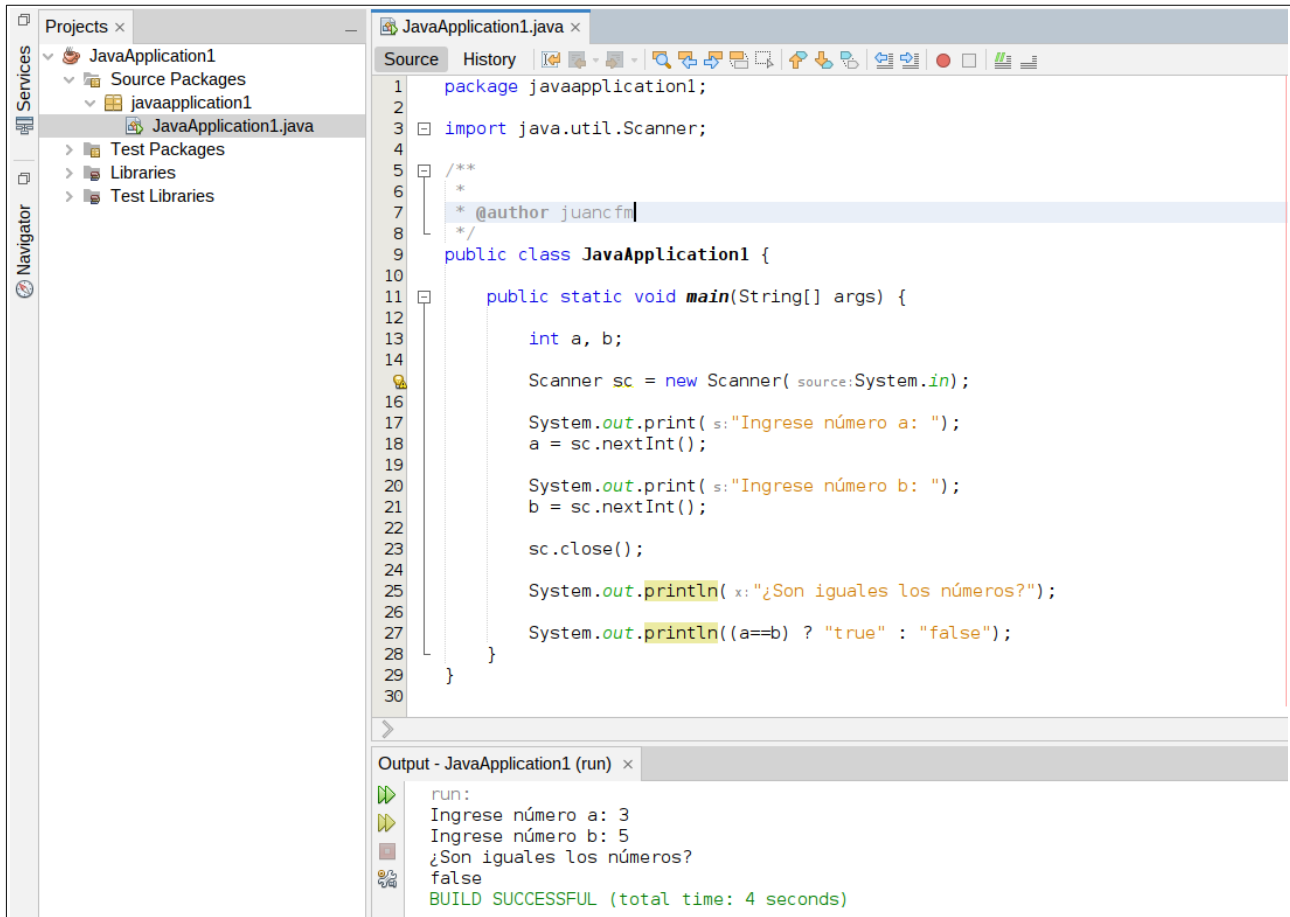
Output - JavaApplication1 (run) ×

```
run:
Ingrese un número: 35
La raíz cuadrada de 35.0 es: 5.916079783099616
BUILD SUCCESSFUL (total time: 5 seconds)
```

[Volver al índice](#)

Actividad 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 project named 'JavaApplication1'. The 'Source' editor displays the following Java code:

```
1 package javaapplication1;
2
3 import java.util.Scanner;
4
5 /**
6  *
7  * @author juancfm
8  */
9 public class JavaApplication1 {
10
11     public static void main(String[] args) {
12
13         int a, b;
14
15         Scanner sc = new Scanner( source: System.in);
16
17         System.out.print( s: "Ingrese número a: ");
18         a = sc.nextInt();
19
20         System.out.print( s: "Ingrese número b: ");
21         b = sc.nextInt();
22
23         sc.close();
24
25         System.out.println( x: "¿Son iguales los números?");
26
27         System.out.println((a==b) ? "true" : "false");
28     }
29 }
30
```

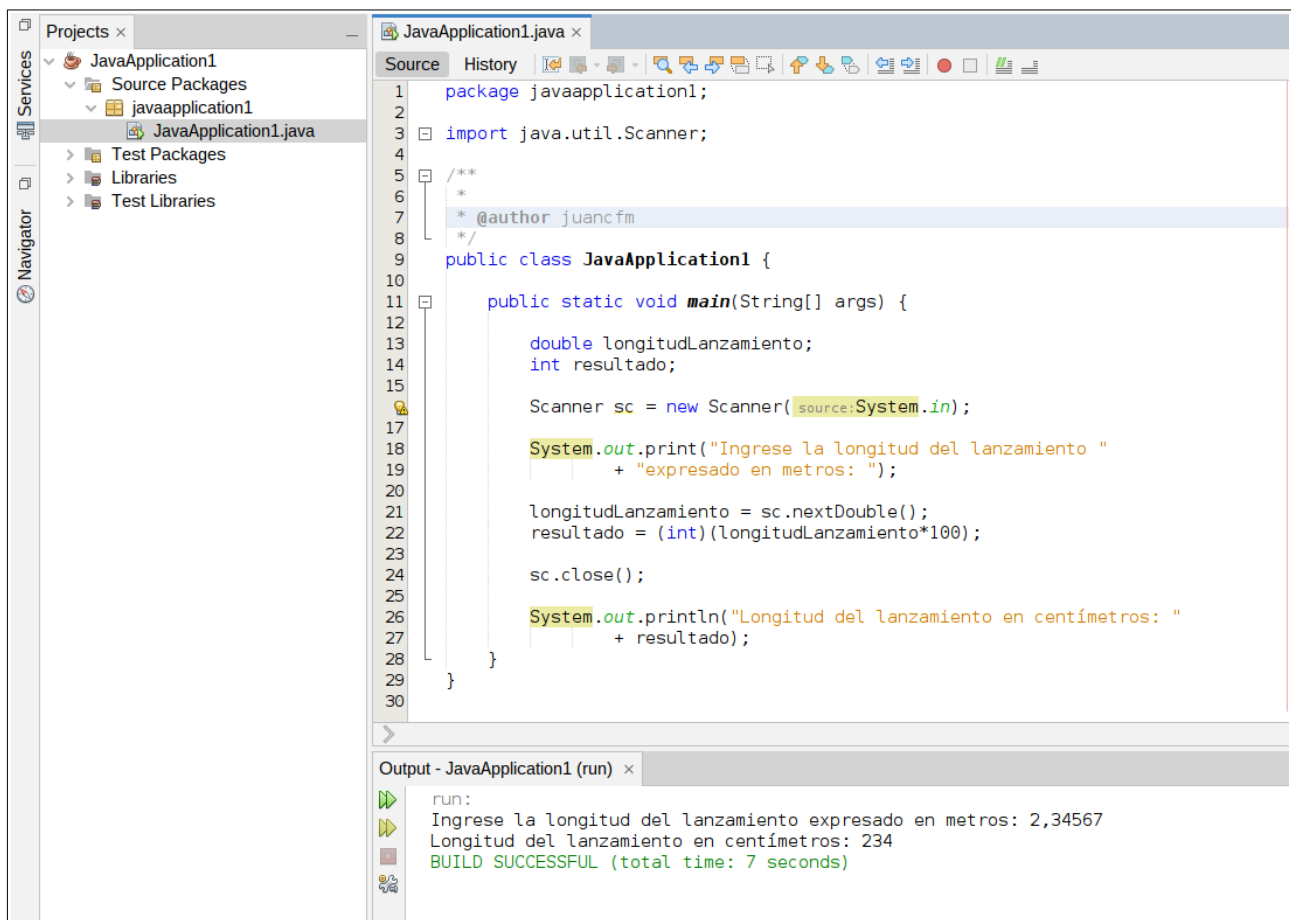
The 'Output - JavaApplication1 (run)' window shows the execution results:

```
run:
Ingrese número a: 3
Ingrese número b: 5
¿Son iguales los números?
false
BUILD SUCCESSFUL (total time: 4 seconds)
```

[Volver al índice](#)

Actividad 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 javaapplication1;
2
3 import java.util.Scanner;
4
5 /**
6  * @author juancfm
7  */
8
9 public class JavaApplication1 {
10
11     public static void main(String[] args) {
12
13         double longitudLanzamiento;
14         int resultado;
15
16         Scanner sc = new Scanner(System.in);
17
18         System.out.print("Ingrese la longitud del lanzamiento "
19             + "expresado en metros: ");
20
21         longitudLanzamiento = sc.nextDouble();
22         resultado = (int)(longitudLanzamiento*100);
23
24         sc.close();
25
26         System.out.println("Longitud del lanzamiento en centímetros: "
27             + resultado);
28     }
29 }
30
```

Output - JavaApplication1 (run) x

```
run:
Ingrese la longitud del lanzamiento expresado en metros: 2,34567
Longitud del lanzamiento en centímetros: 234
BUILD SUCCESSFUL (total time: 7 seconds)
```

[Volver al índice](#)