

# Aprende a programar con Java

## 2.<sup>a</sup> Edición

### Fe de erratas

Página 27. Ejercicio 1.6. Donde pone:

```
* Pediremos tres notas enteras y calcularemos la media. Como la media puede
* tener decimales utilizaremos una variable real. */
```

Debe poner:

```
* Pediremos tres notas enteras y calcularemos la media. Como la media no puede
* tener decimales utilizaremos una variable entera. */
```

Página 116. Epígrafe 5.8.10. Copia. Donde pone:

```
int a[] = Arrays.copyOf(t, 1, 4); // a = [5, 3, 1]
```

Debe poner:

```
int a[] = Arrays.copyOfRange(t, 1, 4); //a = [5, 3, 1]
```

Página 139. Donde pone:

```
int e = 97;
char c = e; //c vale 'a'
```

Debe poner:

```
int e = 97;
char c = (int)e; //c vale 'a'
```

Página 144. Donde pone:

habitualmente, la **primera** forma de asignación.

Debe poner:

habitualmente, la **segunda** forma de asignación.

Página 148. Donde pone:

```
c = "Resultado: " + 'a'); //equivale a "Resultado: " + String.valueOf('a')
```

Debe poner:

```
c = "Resultado: " + 'a'; //equivale a "Resultado: " + String.valueOf('a')
```

Página 149. Donde pone:

`String substring(int inicio)`: devuelve la subcadena formada desde la posición inicio hasta el final de la cadena. Lo que se devuelve es una copia y la cadena invocante no

Debe poner:

`String substring(int inicio)`: devuelve la subcadena formada desde la posición inicio, **inclusive**, hasta el final de la cadena. Lo que se devuelve es una copia y la cadena in-

Página 153. En el epígrafe 6.3.8. Donde pone:

```
cad2 = cad.toUpperCase(); //"mi perro: se llama perico23."  
cad2 = cad.toUpperCase(); //"MI PERRO: SE LLAMA PERICO23."
```

Debe poner:

```
cad2 = cad1.toLowerCase(); //"mi perro: se llama perico23."  
cad2 = cad1.toUpperCase(); //"MI PERRO: SE LLAMA PERICO23."
```

Página 154. Donde pone:

```
cad = String.valueOf(c); //cad vale "Hola"
```

Debe poner:

```
cad = String.valueOf(c); //cad vale "Hola"
```

Página 155. Donde pone:

```
for (int i = 0; i < t.length; i++) { //c = ['0', '1', '2'..., '9']
```

Debe poner:

```
for (int i = 0; i < t.length; i++) { //t = ['0', '1', '2'..., '9']
```

Página 155. Donde pone:

```
//c = ['0', '1', '2', '3', 'u', 'n', '6', '7', '8', '9' ]
```

Debe poner:

```
//t = ['0', '1', '2', '3', 'u', 'n', '6', '7', '8', '9' ]
```

Página 164. Ejercicio 13. Donde pone:

```
a: 4 veces  
d: 1 vez  
e: 2 veces
```

Debe poner:

```
a: 4 veces  
c: 1 vez  
d: 1 vez  
e: 2 veces
```

Página 192. Ejercicio 7.1. Clase CuentaCorriente. Donde pone:

```
saldo -= cant; //no se permite realizar la operación.  
operacionPosible = true;  
} else {
```

Debe poner:

```
saldo -= cant;  
operacionPosible = true;  
} else { //no se permite realizar la operación.
```

Página 205. Clase `Tren`. Donde pone:

```
if (numVagones < 5) {
```

Debe poner:

```
if (numVagones >= 5) {
```

Página 216. Donde pone:

Dos constructores, uno por defecto que cree la `tabla` con capacidad para 10 números, y otro al que se le pasa como argumento el tamaño inicial de la `tabla`.

Debe poner:

Dos constructores, uno por defecto que cree la `lista` con capacidad para 10 números, y otro al que se le pasa como argumento el tamaño inicial de la `lista`.

Página 219. Ejercicio propuesto 7.2. Donde pone:

`void incrementarDia(int cantidad)`: que incrementa, **si resulta correcto**, la fecha del calendario en el número de días especificados.

`void incrementarMes(int cantidad)`: que incrementa la fecha del calendario en el número de meses especificados, **siempre que la fecha resultante sea correcta**.

Debe poner:

`void incrementarDia(int cantidad)`: que incrementa la fecha del calendario en el número de días especificados.

`void incrementarMes(int cantidad)`: que incrementa la fecha del calendario en el número de meses especificados.

Página 222. Donde pone:

donde cada clase tiene como padre una única superclase, cosa no impide que, a su vez, tenga

Debe poner:

donde cada clase tiene como padre una única superclase, cosa **que** no impide que, a su vez,

Página 227. Donde pone:

un objeto `de Persona`, ya que posee todos los miembros de `Persona` —además de otros

Debe poner:

un objeto `Persona`, ya que posee todos los miembros de `Persona` —además de otros

Página 233. Ejercicio 8.1. Donde pone:

`setMinutos(valor)`, que asigna un valor, **si tiene sentido**, a los minutos.

Debe poner:

`setMinutos(valor)`, que asigna un valor, **entre 0 y 59**, a los minutos.

Página 233. Donde pone:

`setHora(valor)`, que asigna un valor, **si tiene sentido**, a la hora.

Debe poner:

`setHora(valor)`, que asigna un valor, **entre 0 y 23**, a la hora.

Página 234. Ejercicio 8.2. Clase `Hora12`. Donde pone:

`//la hora puede llegar a ser las 12:00 (que aquí no tiene sentido)`

Debe poner:

`//la hora puede llegar a ser las 13:00 (que aquí no tiene sentido)`

Página 235. Ejercicio 8.3. Donde pone:

`setSegundo(valor)`, que asigna, **cuando es posible**, el valor indicado a los segun-

Debe poner:

`setSegundo(valor)`, que asigna el valor indicado, **entre 0 y 59, al atributo segundos**.

Página 235. Ejercicio 8.3. Clase `HoraExacta`. Donde pone:

`if (segundos > 60) { //si los segundos son mayores que 60`

Debe poner:

`if (segundos > 59) { //si los segundos son mayores que 59`

Página 241. Donde pone:

`super(ancho, alto, fondo, CajaCarton.Unidad.CM); //reutilizamos constructor`

Debe poner:

`super(ancho, alto, fondo, Unidad.CM); //reutilizamos constructor`

Página 252. Donde pone:

`public tipo metodoAbstracto1(listaPar1) {`

Debe poner:

`public tipo metodo1(listaPar1) {`

Página 273. Donde pone:

crea un flujo de texto asociado al archivo `prueba.txt`, que se halla en la carpeta `programa` de

Debe poner:

crea un flujo de texto asociado al archivo `prueba.txt`, que se halla en la carpeta `programas` de

Página 273. Donde pone:

abre un archivo, **el** cursor se posiciona al principio, apuntando al primer carácter.

Debe poner:

abre un archivo, **un** cursor se posiciona al principio, apuntando al primer carácter.

Página 273. Donde pone:

`int read()`: que lee y devuelve del fichero. El carácter (en *Unicode*) lo devuelve como

Debe poner:

`int read()`: que lee y devuelve del fichero **un carácter**. El carácter (en *Unicode*) lo devuelve

Página 298. Ejercicio 11.1. Donde pone:

```
import java.util.Scanner;
```

```
/*
```

```
 * Crearemos un flujo FileOutputStream de entrada de bytes planos y, a partir
```

Debe poner:

```
import java.util.Scanner;
```

```
import java.util.Locale;
```

```
/* Crearemos un flujo FileOutputStream de entrada de bytes planos y, a partir
```

Página 298. Ejercicio 11.1. Donde pone:

```
Scanner sc = new Scanner(System.in);
```

Debe poner:

```
Scanner sc = new Scanner(System.in).useLocale(Locale.US);
```

Página 313. Donde pone:

A menudo necesitamos guardar información, pero **que** no sabemos de antemano el

Debe poner:

A menudo necesitamos guardar información, pero no sabemos de antemano el

Página 315. Nota a pie de página. Donde pone:

Las clases envoltorio se tratarán en el Anexo B.

Debe poner:

Las clases envoltorio se tratarán en el Anexo B.

Página 316. Donde pone:

`boolean remove(Object ob)`: elimina **un nodo** ob de una lista. Devuelve `true` si la elimi-

Debe poner:

`boolean remove(Object ob)`: elimina **la primera ocurrencia de** ob de una lista. Devuelve `true`

Página 330. Donde pone:

```
Collection<Integer> col = new ArrayList<>();  
lista.add(5);
```

Debe poner:

```
Collection<Integer> col = new ArrayList<>();  
col.add(5);
```

Página 359. Donde pone:

debe ser igual o una superclase de T para garantizar que no va a contener ni evaluar más

Debe poner:

debe ser igual o una superclase de T para garantizar que no va a contener ni evaluar más

Página 361. Nota al pie de página. Donde pone:

pasando es la referencia del bloque donde está su código.

Debe poner:

pasando es la referencia del bloque de memoria donde está su código.

Página 362. Donde pone:

A partir de una tabla: llamando al método of(), de la clase Stream, con la tabla como

Debe poner:

A partir de una tabla: llamando al método of(), de la interfaz Stream, con la tabla como

Página 378. Donde pone:

```
double fin = System.nanoTime();
```

Debe poner:

```
long fin = System.nanoTime();
```

Página 384. Donde pone:

que desenvuelve el 3 de su envoltorio antes de asignarlo a la variable entera w.

Debe poner:

que desenvuelve el 3 de su envoltorio antes de asignarlo a la variable entera int w.