

## Ejercicios CODEGYM nivel 1

Solemnemente publicamos y declaramos estas variables...

VerifyHelpDiscuss

ConditionsClass treeSolution.java

Declara las variables `int` llamadas `a` y `b`.  
En la misma línea en que declares las variables, asignales inmediatamente distintos valores.  
Esos valores pueden ser cualquier **número entero**.

**Requirements:**

- ✓ Declara dos variables `int`.
- ✓ Las variables deben llamarse `a` y `b`.
- ✓ Debes asignar inmediatamente los valores a las variables.
- ✓ Las variables `a` y `b` deben tener valores diferentes.

```
1 package es.codegym.task.task01.task0109;
2
3 /*
4  * Solemnemente publicamos y declaramos estas variables...
5  */
6
7 public class Solution {
8     public static void main(String[] args) {
9         int a = 10;
10        int b = 5;
11    }
12 }
13
```

Una variable en nuestra pantalla

VerifyHelpDiscuss

ConditionsClass treeSolution.java

Crea una variable de tipo `String` llamada `nombre`.  
En la misma línea donde declares la variable, asignale inmediatamente algún valor.  
Muestra la variable `nombre` en la pantalla.

**Requirements:**

- ✓ En el método `main`, declara una variable de tipo `String`.
- ✓ La variable debe llamarse `nombre`.
- ✓ Debe asignarse un valor a la variable de inmediato.
- ✓ El programa debe mostrar texto en la pantalla.
- ✓ El texto de salida debe ser el valor de la variable.

```
1 package es.codegym.task.task01.task0110;
2
3 /*
4  * Una variable en nuestra pantalla
5  */
6
7 public class Solution {
8     public static void main(String[] args) {
9         String nombre = "CodeGym";
10        System.out.println(nombre);
11    }
12 }
13
```

Output

CodeGym

Una variable no es suficiente

Verify

Help

Discuss

Conditions

Class tree

Solution.java

Declara las variables `int a` y `b` y la variable de tipo `String s`.  
En la misma línea en que declares las variables, asignales inmediatamente distintos valores.

**Requirements:**

- ✓ Declara dos variables `int`.
- ✓ Declara una variable `String`.
- ✓ Las variables `int` deben llamarse `a` y `b`.
- ✓ La variable `String` debe llamarse `s`.
- ✓ Debes asignar inmediatamente los valores a las variables.
- ✓ Las variables `a` y `b` deben tener valores diferentes.

```
1 package es.codegym.task.task01.task0111;
2
3 /*
4  * Una variable no es suficiente
5  */
6
7 public class Solution {
8     public static void main(String[] args) {
9         int a = 5;
10        int b = 10;
11        String s = "CodeGym";
12    }
13 }
14
```

Mostrar el cuadrado de un número

Verify

Help

Discuss

Conditions

Class tree

Solution.java

Se proporciona la variable `numero`.  
Escribe un programa que **muestre el cuadrado de esta variable**: (`numero * numero`).

**Requirements:**

- ✓ No cambies la variable `numero`. Solo puedes cambiar su valor.
- ✓ El programa debe generar texto.
- ✓ El texto mostrado debe ser un número.
- ✓ El texto mostrado debe ser el cuadrado de la variable `numero`.

```
1 package es.codegym.task.task01.task0140;
2
3 /*
4  * Mostrar el cuadrado de un número
5  */
6
7 public class Solution {
8     public static int numero = 25;
9
10    public static void main(String[] args) {
11        System.out.println(numero * numero);
12    }
13 }
```

Output

625

ConditionsClass treeSolution.java

Escribe un programa que muestre: «CodeGym: apréndelo una vez y úsalo en cualquier lugar» diez veces.

Requirements:

- El programa debe generar texto.
- El texto debe comenzar con «CodeGym».
- El texto debe terminar con «úsalo en cualquier lugar».
- El texto debe constar de diez líneas.
- El texto mostrado debe cumplir las condiciones de la tarea.

```
1 package es.codegym.task.task01.task0114;
2
3 /*
4  * CodeGym: apréndelo una vez y úsalo en cualquier lugar
5  */
6
7 public class Solution {
8     public static void main(String[] args) {
9         String texto = "CodeGym: apréndelo una vez y úsalo en cualquier lugar";
10
11         System.out.println(texto);
12         System.out.println(texto);
13         System.out.println(texto);
14         System.out.println(texto);
15         System.out.println(texto);
16
17         System.out.println(texto);
18         System.out.println(texto);
19         System.out.println(texto);
20         System.out.println(texto);
21         System.out.println(texto);
22     }
23 }
24
```

Output

CodeGym: apréndelo una vez y úsalo en cualquier lugar  
CodeGym: apréndelo una vez y úsalo en cualquier lugar  
CodeGym: apréndelo una vez y úsalo en cualquier lugar  
CodeGym: apréndelo una vez y úsalo en cualquier lugar  
CodeGym: apréndelo una vez y úsalo en cualquier lugar  
CodeGym: apréndelo una vez y úsalo en cualquier lugar  
CodeGym: apréndelo una vez y úsalo en cualquier lugar  
CodeGym: apréndelo una vez y úsalo en cualquier lugar  
CodeGym: apréndelo una vez y úsalo en cualquier lugar  
CodeGym: apréndelo una vez y úsalo en cualquier lugar

Eh... ¡Feliz año nuevo!

Verify

Help

Discuss

ConditionsClass treeSolution.java

Elimina los comentarios de parte del código para que aparezca «Feliz Año Nuevo» en la pantalla.

Requirements:

- El programa debe mostrar texto en la pantalla.
- Tienes que quitar el comentario de algunas líneas. No cambies nada más.
- No añadas nuevas líneas al programa ni elimines las existentes.
- El texto mostrado debe cumplir las condiciones de la tarea.

```
1 package es.codegym.task.task01.task0115;
2
3 /*
4  * Eh... ¡Feliz año nuevo!
5  */
6
7 public class Solution {
8     public static void main(String[] args) {
9         String s = "Feliz";
10         System.out.println("Navidad");
11         System.out.println("Feliz Nuevo");
12         System.out.print("Fin de semana");
13         System.out.print("Feliz Año");
14         System.out.print("Fin de semana");
15         System.out.println(s);
16         System.out.print("¡Feliz Nuevo!");
17         System.out.print(" ");
18         System.out.println("Nuevo");
19     }
20 }
21
```

Output

Feliz Año Nuevo

ConditionsClass tree

Solution.java

Cambia el programa para que la variable **nombre** adopte el valor «Amigo».

Requirements:

- A la variable nombre se le debe asignar un valor inmediatamente después de su declaración.
- Se deben declarar dos variables.
- El valor de la variable texto no se puede cambiar.
- No puedes cambiar el comando de salida por pantalla.
- El programa debe mostrar «¡Hola, Amigo!».

```
1 package es.codegym.task.task01.task0116;
2
3 /*
4  * Combiemos el código
5  */
6
7 public class Solution {
8     public static void main(String[] args) {
9         String nombre = "Amigo";
10        String texto = "¡Hola, " + nombre + "!";
11        System.out.println(texto);
12    }
13 }
14
```

Output

¡Hola, Amigo!

Algunas modificaciones más

VerifyHelpDiscuss

ConditionsClass tree

Solution.java

Cambia el programa para que muestre el siguiente texto:

Codificación en Java

Usa variables.

Requirements:

- Se deben declarar tres variables.
- A todas las variables se les debe asignar un valor inmediatamente después de declararlas.
- No puedes cambiar el comando de salida por pantalla.
- El programa debe mostrar «Codificación en Java».

```
1 package es.codegym.task.task01.task0117;
2
3 /*
4  * Algunas modificaciones más
5  */
6
7 public class Solution {
8     public static void main(String[] args) {
9         String s1 = "Codificación";
10        String s2 = "en";
11        String s3 = "Java";
12        System.out.println(s1 + " " + s2 + " " + s3);
13    }
14 }
15
```

Output

Codificación en Java

Comenta lo superfluo

Verify

Help

Discuss

Conditions

Class tree

Solution.java

Comenta algo de código para que aparezca «suma = 12» en la pantalla.

**Requirements:**

- El programa debe mostrar texto en la pantalla.
- Debes comentar las líneas necesarias y dejar sin modificar las demás.
- No añadas nuevas líneas al programa ni elimines las existentes.
- El texto mostrado debe cumplir las condiciones de la tarea.

```
1 package es.codegym.task.task01.task0118;
2
3 /*
4  * Comenta lo superfluo
5  */
6
7 public class Solution {
8     public static void main(String[] args) {
9         int a = 3;
10        //int a = 10;
11        int b = 6;
12        //int b = 12;
13        //int suma = 1 + a + b;
14        //int suma = 2 + a + b;
15        int suma = 3 + a + b;
16        //int suma = 4 + a + b;
17
18        System.out.println("suma = " + suma);
19    }
20 }
21
```

Output

suma = 12

¡Vuelvo a tener quince años!

Verify

Help

Discuss

Conditions

Class tree

Solution.java

Cambia el programa para que la variable **edad** adopte el valor **15**.

**Requirements:**

- Se deben declarar cuatro variables.
- La variable edad debe ser un int.
- A todas las variables se les debe asignar un valor inmediatamente después de declararlas.
- El valor asignado a la variable edad debe ser 15.

```
1 package es.codegym.task.task01.task0119;
2
3 /*
4  * ¡Vuelvo a tener quince años!
5  */
6
7 public class Solution {
8     public static void main(String[] args) {
9         String nombre = "Anigo";
10        int edad = 15;
11        int dinero = 500;
12        int peso = 70;
13    }
14 }
15
```

Solo valdrá si da 20

VerifyHelpDiscuss

ConditionsClass treeSolution.java

En el método `main`, pon correctamente los **signos más y menos** para que la variable `resultado` dé 20.

Solo debes colocar los signos en la línea donde se declara la variable `resultado`.

No cambies el orden de las variables en esta línea.

Cada variable debe ir precedida por un signo **más** o **menos**.

**Requirements:**

- ✓ Valores de las variables: no cambies a, b, c ni d.
- ✓ Cada una de las variables (a, b, c y d) de la línea donde se declara la variable resultado debe ir precedida por un signo más o menos.
- ✓ El programa debe mostrar en la pantalla el número 20.
- ✓ Los signos más y menos deben colocarse correctamente.

```
1 package es.codegym.task.task01.task0137;
2
3 /*
4  Solo valdrá si da 20
5  */
6
7 public class Solution {
8     public static int a = 1;
9     public static int b = 3;
10    public static int c = 9;
11    public static int d = 27;
12
13    public static void main(String[] args) {
14
15        int resultado = - a + b - c + d;
16
17        System.out.println(resultado);
18    }
19 }
```

Output

20

Bésame mi brillante retropropulsor

VerifyHelpDiscuss

ConditionsClass treeSolution.java

- «Amigo, ¿quieres ser programador?».

- «¡Desde luego!».

- «Entonces escribe rápidamente un programa que muestre la frase *¡Hola mundo!*».

- «Pero... ¡eso está muy visto!».

- «Así es. Amigo, veo que piensas como yo, y eso es genial. Mejor vamos a hacer lo siguiente: muestra una frase que nosotros, como robots, deberíamos repetir en estos tiempos de incertidumbre. La frase es *¡Bésame mi brillante retropropulsor!*».

**Requirements:**

- ✓ El programa debe generar texto.
- ✓ El texto debe comenzar con «¡Bésame!».
- ✓ El texto debe terminar con un signo de exclamación.
- ✓ El texto debe constar de 36 caracteres, incluidos espacios y signos de puntuación.
- ✓ El texto mostrado debe cumplir las condiciones de la tarea.

```
1 package es.codegym.task.task01.task0139;
2
3 /*
4  Bésame mi brillante retropropulsor
5  */
6
7 public class Solution {
8     public static void main(String[] args) {
9         System.out.println("¡Bésame mi brillante retropropulsor!");
10    }
11 }
```

Output

¡Bésame mi brillante retropropulsor!

Contrato

Verify

Help

Discuss

Conditions

Class tree

Solution.java

Muestra el texto del nuevo contrato:

Me llamo Amigo.

Acepto un salario de 5000 créditos al mes el primer año.

Acepto un salario de 5500 créditos al mes el segundo año.

Acepto un salario de 7000 créditos al mes el tercer año.

Acepto un salario de 8000 créditos al mes el cuarto año.

Acepto un salario de 10 000 créditos al mes el quinto año.

¡Bésame mi brillante retropropulsor!

Requirements:

✓ El programa debe generar texto.

✓ El texto debe comenzar con «Me llamo Amigo».

✓ El texto debe terminar con «brillante retropropulsor!».

✓ El texto debe constar de nueve líneas (las vacías también cuentan).

1package es.codegym.task.task01.task0121;

2

3/\*

4Contrato

5\*/

6

7public class Solution {

8public static void main(String[] args) {

9System.out.println("Me llamo Amigo.");

10System.out.println("");

11System.out.println("Acepto un salario de 5000 créditos al mes el primer año

12System.out.println("Acepto un salario de 5500 créditos al mes el segundo año

13System.out.println("Acepto un salario de 7000 créditos al mes el tercer año

14System.out.println("Acepto un salario de 8000 créditos al mes el cuarto año

15System.out.println("Acepto un salario de 10 000 créditos al mes el quinto año

16System.out.println("");

17System.out.println("¡Bésame mi brillante retropropulsor!");

18}

19}

20}

Output

Me llamo Amigo.

Acepto un salario de 5000 créditos al mes el primer año.

Acepto un salario de 5500 créditos al mes el segundo año.

Acepto un salario de 7000 créditos al mes el tercer año.

Acepto un salario de 8000 créditos al mes el cuarto año.

Acepto un salario de 10 000 créditos al mes el quinto año.

Encontrar errores

Verify

Help

Discuss

Conditions

Class tree

Solution.java

Arregla el error en el programa. El programa debería mostrar el número 100.

Requirements:

✓ El programa debe mostrar texto en la pantalla.

✓ Se deben declarar tres variables.

✓ No cambies los tipos de las variables.

✓ No cambies los valores de las variables.

✓ No añadas nuevas líneas al programa ni elimines las existentes.

✓ Cambia lo menos posible el comando de salida por pantalla.

✓ El texto mostrado debe cumplir las condiciones de la tarea.

1package es.codegym.task.task01.task0122;

2

3/\*

4Encontrar errores

5\*/

6

7public class Solution {

8public static void main(String[] args) {

9int a = 56;

10int b = 30;

11int c = 14;

12System.out.print(a + b + c);

13}

14}

15}

Output

100

Conditions

Class tree

4

Solution.java

Comenta las variables que no se usen. El programa debe compilar.


**Requirements:**


- ✓ No cambies los tipos de las variables.
- ✓ No cambies los valores de las variables.
- ✓ No añadas nuevas líneas al programa ni elimines las existentes.
- ✓ Tienes que comentar las variables que no se utilicen.
- ✓ No debes cambiar la salida del programa.

```
1 package es.codegym.task.task01.task0123;
2
3 /*
4  * No hacen falta líneas de más
5  */
6
7 public class Solution {
8     public static void main(String[] args) {
9         int a = 10;
10        int b = 15;
11        double c = b + 38;
12        //int d = a + 12;
13        //double e = 12.3;
14        String s = "s" + a;
15        String s1 = a + "b";
16        //String s2 = "a";
17        String s3 = s1 + "a";
18        String s4 = s3 + "b";
19        System.out.println(c + s4 + s);
20    }
21 }
22
```

Output

53.010babs10



Task  Sintaxis de Java, level 1, lesson 7

COMPLETED


EASY


★★★★★

### Entrada de código

A veces no necesitas pensar, ¡solo tienes que teclearlo! Por paradójico que parezca, en ocasiones tus dedos «recuerdan» mejor que tu mente consciente. Por eso, durante tu formación en el centro secreto CodeGym, encontrarás algunas tareas que requieren que introduzcas código. Al hacerlo, te acostumbrarás a la sintaxis y ganarás un poco de materia oscura. Además, ¡combatirás la pereza!

Open



Task  Sintaxis de Java, level 1, lesson 8

COMPLETED


EASY

### La gran purga

Quien haya escrito este programa, obviamente, lo hizo a toda prisa. Bueno, eso no es cierto: este programa se ha escrito con fines educativos y el autor metió adrede variables superfluas al tiempo que dejó sin declarar las necesarias. Vamos a corregirlo: Comenta las variables innecesarias y declara las que falten. Solo entonces alcanzará el programa la armonía universal.

Open



Task  Sintaxis de Java, level 1, lesson 8


COMPLETED


EASY

### ¿No te apetece? Hazlo de todas formas

La pereza aflige incluso a los mejores programadores. Y no solo a los programadores. No obstante, las personas logran convertirse en profesionales enseñándose a sí mismas. Por eso te sugerimos que no seas perezoso. Muestra este eslogan en la pantalla: «Si te apetece, haz la tarea; si no, también». Y para no olvidarlo, imprímelo dieciséis veces.

Open



Task  Sintaxis de Java, level 1, lesson 8


COMPLETED

EPIC

### Cuadrado de un número

Hay varias formas de obtener el cuadrado de un número. Por ejemplo, hay quienes escriben el número y, luego, dibujan un cuadrado a su alrededor. Se trata de un método que emplean las personas que no han estudiado. Los demás toman sus multiplicaciones de la tabla de multiplicar y usan un programa. Tu programa debe mostrar el





8

x5

Task

Sintaxis de Java, level 1, lesson 8

COMPLETED

EPIC

### Tan fácil como sumar 2 + 2

En una aplicación Java, todas las acciones las llevan a cabo las funciones; más exactamente, los métodos. En nuestro programa, un amable mentor ya ha implementado un método (es decir, escrito el código del método) que puede calcular la suma de dos números. Lo único que tienes que hacer es llamar a ese método con los argumentos 2 y 2. Has de hacerlo desde el método main.

Open

{



8

x1

Task

Sintaxis de Java, level 1, lesson 8

COMPLETED


EASY

### Aliméntate bien: ¡come fruta!

Muestra las variables cuyos valores son nombres de frutas. Muestra cada variable en una nueva línea.

Open

{



8

x1

Task

Sintaxis de Java, level 1, lesson 8

COMPLETED

EASY

### Mi poema favorito

Los programadores de hoy en día rara vez son ingenieros de pleno derecho obsesionados con la tecnología, los dispositivos y nada más. También los hay apasionados de la poesía. Si eres uno de esos amantes de la poesía, te resultará fácil realizar la siguiente tarea: muestra un cuarteto (o varios) de tu poema favorito.

Open

{