

Segundo Ejercicio Práctico

https://ocw.uc3m.es/pluginfile.php/2379/mod_page/content/12/practica_java_basics.pdf

Este ejercicio práctico consta de tres partes:

- La primera parte es una introducción al entorno de programación con Eclipse. No se entregará ningún material para esta parte.
- La **segunda parte tiene como objetivo introducir la programación en Java**. Esta parte incluye una serie de ejercicios básicos que los estudiantes deberán desarrollar, así como un conjunto de preguntas asociadas a cada ejercicio. No se entregará el código de estos ejercicios.
- La tercera parte incluye ejercicios más complejos. El código de estos ejercicios deberá entregarse antes del 08 de mayo de 2026.

2. Segunda parte: Introducción a la programación Java

La segunda parte tiene como objetivo aplicar los conocimientos adquiridos. Para ello, se proponen diversos ejercicios de dificultad creciente.

2.1. Tipos de datos, arreglos y operadores básicos de Java

Algunos de los siguientes ejercicios contienen errores. En estos casos, el informe del estudiante debe explicar cuáles son estos errores y por qué ocurren. Para continuar la ejecución del programa incorrecto, una vez localizados y explicados los errores, agregue dos barras diagonales (//) antes de la línea que produce el error. De esta manera, Java interpretará esta línea como un comentario y el error se ignorará.

Nota: No entregue el código de los ejercicios de esta sección. En su lugar, los estudiantes deben responder las preguntas en sus informes y completar la prueba en Aula Global (fecha por determinar). Los ejercicios deben desarrollarse con el entorno Eclipse.

Ejercicio 1. Declaración de variables

Cree una clase llamada `Ejercicio1`. Dentro de esta clase, cree un método `main` y, dentro de este método, declare ocho variables, cada una con un tipo de dato básico de Java diferente (una declaración por oración). Declare una variable de tipo `String`. A continuación, asigne valores válidos a cada variable. Imprima los valores de las variables en pantalla con las instrucciones `System.out.println(<nombre de la variable>)`, reemplazando `<nombre de la variable>` por el nombre de cada variable.

Ejercicio 2. Rango de tipos de datos básicos

Continúe con el ejercicio anterior (sin modificar nada, simplemente agregue las nuevas instrucciones debajo de la última instrucción `System.out.println`) agregando instrucciones

de asignación para las variables numéricas con valores fuera del rango de cada tipo. Imprima los resultados en pantalla.

¿Qué sucede en cada caso?

Ejercicio 3. Precisión de tipos numéricos

Continúe con el ejercicio anterior (sin modificar nada, simplemente agregue las nuevas instrucciones debajo de la última instrucción `System.out.println`) declarando dos variables de tipo `float`. Inicialice la primera variable con el valor `1234567890F` y la segunda con el valor `1234567899F`. ¿Cuál es el resultado? Repita el mismo procedimiento con dos variables enteras. ¿Cuál es el resultado?

Ejercicio 4. Uso de variables

Cree una clase llamada `Ejercicio4`. Dentro de esta clase, cree un método `main` y, dentro de este método, declare una variable (de cualquier tipo) sin valor inicial. Imprima esta variable en pantalla antes de asignarle cualquier valor. ¿Qué sucede? ¿Por qué?

Cree una segunda variable del mismo tipo y asígnele el valor de la primera variable. ¿Qué sucede? ¿Por qué?

Ejercicio 5. Declaraciones múltiples

Cree una clase llamada `Ejercicio5`. Dentro de esta clase, cree un método `main` y, dentro de este método, declare en una sola línea 3 variables del mismo tipo básico (cualquiera de ellas).

Declara en una sola frase tres variables de un tipo diferente a la anterior, asignando un valor a la primera y a la última en la misma declaración. A continuación, en otra frase, asigna un valor a la segunda variable. Declara una variable del mismo tipo y asígnele un valor resultante de la combinación de ambas (por ejemplo, la suma de todas ellas). Imprime las cuatro variables en pantalla.

Añade a continuación una instrucción que modifique el valor de la primera variable de este segundo conjunto. Imprime el valor de la cuarta variable. ¿Cambia el valor de la cuarta variable (la variable “combinada”) tras modificar la primera? ¿Por qué?

Ejercicio 6. Constantes

Continúa el ejercicio anterior añadiendo `final` antes de la declaración del segundo conjunto de variables. ¿Qué ocurre? ¿Por qué?

Elimina el modificador `final` anterior y añade `final` antes de la declaración de la cuarta variable. ¿Qué ocurre? ¿Por qué?

Crea una constante de tipo `String` y asígnele un valor. En la siguiente línea, modifica su valor. ¿Es posible?

Ejercicio 7. Cadenas de texto (tipo de dato ‘char’)

Crea una clase llamada `Ejercicio7`. Dentro, crea un método `main` y, dentro de este método, escribe el siguiente código:

```
char a;  
a = '\\\\';  
System.out.println(a);
```

¿Cuál es el resultado de la ejecución de las oraciones anteriores? ¿Por qué?

¿Es posible crear un carácter con más de un carácter (por ejemplo, `char b = 'hola'`)?

Ejercicio 8. Copia de variables

Creas dos variables de cualquier tipo básico, asigna un valor a la primera y haz que la segunda sea igual a la primera. Imprime en pantalla el valor de la segunda variable. Amplía el programa con una nueva instrucción que cambie el valor de la primera variable y añade otra instrucción para imprimir de nuevo el valor de la segunda. ¿Cambia el valor de la segunda variable? ¿Por qué? Repite el mismo procedimiento con dos variables de tipo `String`.

Ejercicio 9. Declaración de arreglos (I)

Creas una clase llamada `Ejercicio9`. Dentro de ella, crea un método `main`. Dentro de este método, declara un arreglo de cualquier tipo de dato entero, un arreglo de cualquier tipo de dato real, un arreglo de `char`, un arreglo `boolean` y un arreglo de `String`. Crea estos arreglos con diferentes tamaños. En las siguientes líneas, asigna valores a algunos de los elementos de los arreglos. ¿Qué sucede cuando usamos (por ejemplo, imprimimos) algunos de los elementos que no se han inicializado? ¿Cuál es la diferencia con respecto al ejercicio 3?

Creas un arreglo de cualquier tipo proporcionando un valor inicial para sus elementos en la declaración.

Usa `length` para imprimir en pantalla el tamaño de los arreglos anteriores. Cambia la longitud de uno de estos arreglos escribiendo `array.length = 5`. ¿Qué sucede? ¿Por qué?

Selecciona uno de los arreglos anteriores e iguala dos de sus elementos (por ejemplo, `a[5] = a[3]`). Imprime ambos elementos. Luego, agrega una línea que cambie el valor del segundo elemento e imprime ambos elementos nuevamente. ¿Cambia también el primero? ¿Por qué?

Ejercicio 10. Copia de un array

Creas una clase llamada `Ejercicio10`. Dentro, crea un método `main`. Crea un array de cualquier tipo. Asigna un valor a cada uno de sus elementos. Crea un segundo array e iguala su valor al primero (con `a = b`). Modifica el valor de un elemento del primer array. ¿Cambia el valor del elemento correspondiente del segundo array? ¿Por qué?

Creas dos arrays adicionales y repite los pasos anteriores, pero en lugar de `=`, usa

```
System.arraycopy(origen, primer elemento, destino, primer elemento,
número de elementos)
```

¿Notas alguna diferencia?

Ejercicio 11. Arrays multidimensionales

Creas una clase llamada `Ejercicio11`. Dentro, crea un método `main`. Crea un array bidimensional de cualquier tipo y asigna valores a sus elementos en la declaración.

Creas un array bidimensional de un tipo diferente. Asigna valores a sus elementos uno por uno.

Ejercicio 12. Longitud de las dimensiones de un array

Creas una clase llamada `Ejercicio12` y su método `main` correspondiente. Define un arreglo de `String` de `3x4` y asigna valores a sus elementos. Imprime en pantalla el número de

filas y columnas del arreglo usando la propiedad `length`.

Ejercicio 13. Operaciones aritméticas con enteros (I)

Crea una clase llamada `Ejercicio13` y su método `main` correspondiente. Escribe el siguiente código:

```
int a;
int b;
int c;
int d;
a = 2;
b = 3 * 3;
c = 7 / 3;
d = a + b * c;
System.out.println(d);
```

¿Cuál es el resultado del código anterior? ¿Por qué?

Ejercicio 14. Operaciones aritméticas con enteros (II)

Crea una clase llamada `Exercise14` y su método `main` correspondiente. Declara tres variables para cada uno de los cuatro tipos de datos enteros (`byte`, `short`, `int`, `long`) (12 variables) y tres variables de tipo `char`. Para cada tipo (enteros y caracteres), proporciona valores iniciales para las dos primeras variables y asigna a la tercera el resultado de las operaciones con todos los operadores aritméticos de Java (suma, resta, etc.). Imprime los resultados en pantalla.

¿Obtienes algún error? ¿Por qué? (Nota: Una vez detectados los errores, añade `//` al principio de las líneas incorrectas para convertirlas en comentarios, en lugar de borrarlas).

Ejercicio 15. Operaciones aritméticas con enteros (III)

Crea una clase llamada `Exercise15` y su método `main` correspondiente. Declara tres variables de tipo `int`. Asigna el valor 5 a la primera y el valor 0 a la segunda. Asigna a la tercera variable el resultado de dividir la primera variable entre la segunda. Imprime el resultado en pantalla.

¿Hay algún error? ¿Por qué? ¿Cambia el resultado si las variables se declaran con otro tipo entero en lugar de `int`?

Ejercicio 16. Autoincremento

Crea una clase llamada `Ejercicio16` y su método `main` correspondiente. Declara dos variables de tipo `long`, asigna un valor a la primera y haz que la segunda sea igual al autoincremento de la primera (con el operador de autoincremento prefijo). Imprímelas.

A continuación, crea dos variables adicionales y repite el mismo código, pero usando el operador de autoincremento posfijo. Imprime las variables. ¿Hay alguna diferencia? ¿Por qué?

Repite estos pasos para el operador de autodecremento.

Ejercicio 17. Operaciones aritméticas con 'char'

Crea una clase llamada `Ejercicio17` y su método `main` correspondiente. Declara una variable de tipo `char` con un valor inicial adecuado.

Imprímela con `System.out.println(variable+1)`. ¿Qué sucede? ¿Por qué?

Imprímelo con `System.out.println(++variable)`. ¿Qué sucede? ¿Por qué?

Ejercicio 18. Concatenación de cadenas

Crema una clase llamada `Exercise18` y su método `main` correspondiente. Declara tres variables de tipo `String`. Asigna cualquier valor a las dos primeras variables y haz que la tercera sea igual a la suma de la primera y la segunda. Imprime la tercera variable. ¿Qué sucede? ¿Qué sucede si haces que la tercera sea igual a la primera menos la segunda?

Ejercicio 19. Operaciones aritméticas con números reales

Repita los ejercicios 13, 14 y 15, pero usa tipos de datos reales (`double` o `float`) en lugar de `int`. ¿Notas alguna diferencia?

Ejercicio 20. Conversión automática de tipos

Crema una clase llamada `Ejercicio20` y su método `main` correspondiente. Declara y asigna un valor inicial a las variables de cada tipo de dato básico, incluyendo `String`. A continuación, asigna una variable a las demás y repite el proceso para cada una. En tu informe, completa la siguiente tabla con las asignaciones válidas. Escribe "SÍ" si la variable de la fila se puede asignar a la variable de la columna (es decir, `<variable de columna>= <variable de fila>` es posible); de lo contrario, escribe "NO".

Type	byte	short	int	Long	float	double	char	boolean	String
byte	YES								
short		YES							
int			YES						
long				YES					
float					YES				
double						YES			
char							YES		
boolean								YES	
String									YES

Ejercicio 21. Conversión de tipos forzada por el usuario

Crema una clase llamada `Ejercicio21` con el mismo contenido que el `Ejercicio20`. Usa el operador de conversión para forzar la conversión entre todos los tipos y completa la tabla. ¿Quedan celdas con "NO"? Si alguna celda sigue con "NO", explica por qué.

Ejercicio 22. Conversión de tipos

Crema una clase llamada `Ejercicio22` con su método `main` correspondiente. Escribe el siguiente código dentro del método `main`:

```
char a;
int b;
short c;
long d;
float e;
double f;

System.out.println("Asignamos una variable de tipo carácter a los enteros.");
a = '4';
b = a;
c = a;
d = a;
System.out.println("a:" + a);
```

```

System.out.println("b:" + b);
System.out.println("c:" + c);
System.out.println("d:" + d);

System.out.println("Asignamos una variable de tipo double a una variable de tipo float.");
f = 1e200;
e = f;
System.out.println("f:" + f);
System.out.println("e:" + e);

System.out.println("Asignamos una variable flotante a un entero.");
e = 1234.5678;
b = e;
System.out.println("e:" + e);
System.out.println("b:" + b);

```

¿Cuál es el resultado del programa anterior? ¿Por qué?

Ejercicio 23. Operadores relacionales

Crema una clase llamada `Ejercicio23` con su método `main`. Escribe el siguiente código dentro del método `main`:

```

int a,b;
float c=3;
boolean r,s,t,u,v,w,x;
a = 3;
b = 8;
r = a == 0;
s = a != 0;
t = a <= b;
u = b >= a;
v = b > a;
w = b < a;
x = c == 3.0;
System.out.println("r:" + r);
System.out.println("s:" + s);
System.out.println("t:" + t);
System.out.println("u:" + u);
System.out.println("v:" + v);
System.out.println("w:" + w);
System.out.println("x:" + x);

```

¿Cuál es el resultado del programa anterior? ¿Por qué?

Ejercicio 24. Depuración de programas con tipos de datos básicos

Ejecute el programa desarrollado en el ejercicio anterior en modo depuración: utilice un punto de interrupción y la ejecución paso a paso. ¿Puede observar cómo cambian los valores de las variables? ¿Qué ventajas ofrece el depurador en este caso?

Ejercicio 25. Operadores lógicos

Crema una clase llamada `Exercise25` con su correspondiente método `main`. Escriba el siguiente código dentro del método `main`:

```
int a,b;
boolean r,s,t;
a = 3;
b = 8;
r = a == 0 | b >= a;
s = a != 0 & b < a;
t = a <= b ^ b > a;
System.out.println("r:" + r);
System.out.println("s:" + s);
System.out.println("t:" + t);
```

¿Cuál es el resultado del programa anterior? ¿Por qué?

Ejercicio 26. Operadores de asignación

Crea una clase llamada `Ejercicio26` con su correspondiente método `main`. Escribe el siguiente código dentro del método `main`:

```
int a=5,b=3;
boolean r=true,s=false;
a+=b+8*b;
r&=s;
System.out.println("a:" + a);
System.out.println("b:" + b);
System.out.println("r:" + r);
System.out.println("s:" + s);
```

¿Cuál es el resultado del programa anterior? ¿Por qué?

Ejercicio 27. Precedencia de operadores

Crea una clase llamada `Ejercicio27` con su método `main`. Escribe el siguiente código dentro del método `main`:

```
int a=5,b=3,c=20,d=20;
c-=++a/b-3+a%b;
d-=++a/(b+3-4*a)%b;
System.out.println("c:" + c);
System.out.println("d:" + d);
```

¿Cuál es el resultado del programa anterior? ¿Por qué?

Ejercicio 28. Ámbito de las variables

Crea una clase llamada `Ejercicio28` con su método `main`. Dentro del método `main`, declara una variable de cualquier tipo. A continuación, declara la misma variable (mismo nombre y mismo tipo). ¿Es posible? ¿Por qué? ¿Qué ocurre si el *nombre* es el mismo, pero el *tipo* es diferente?

Encierra la declaración de la primera variable entre llaves . ¿Puedes declarar ahora la misma variable fuera de las llaves? ¿Por qué?

Asigna un valor a la variable dentro de las llaves. Imprímelo fuera de las llaves. ¿Qué ocurre? ¿Por qué?

Ejercicio 29. Arreglos irregulares

Crea una clase llamada `Ejercicio29` con su método `main`. Define un arreglo irregular bidimensional de tipo `String` llamado `año`. Cada fila representará un mes y las columnas de cada fila representarán los días del mes (en un año no bisiesto). Es decir, la primera fila debe tener 31 elementos, la segunda 28, etc. Asigne el valor “*no hay clase hoy*” a la celda correspondiente al 8 de enero. Asigne otros valores a la celda correspondiente a la fecha actual. Imprima en pantalla el número de filas y columnas del array utilizando la propiedad `length`.

Ejercicio 30. Argumentos del método ‘main’

Al ejecutar el método ‘main’ de una clase, podemos agregar parámetros de ejecución (consulte el tutorial de Eclipse en la Parte I). Estos parámetros se almacenan en un array de cadenas llamado `args`. Este array, especificado en la declaración del método ‘main’, es creado y asignado automáticamente por Java.

Cree una clase llamada ‘`Exercise30`’ con un método ‘main’ correspondiente que imprima en pantalla los tres primeros argumentos pasados al programa. ¿Qué sucede si se llama al programa con menos de tres argumentos? ¿Y si se llama con más de tres argumentos?

Ejercicio 31. Bucles y arrays

Defina un array de 100 elementos enteros. Utiliza un bucle `while` para inicializar el array con los números del 100 al 199. A continuación, utiliza otro bucle `while` para imprimir en pantalla el contenido del array.

Ejercicio 32. Objetos como estructuras de datos

Define una clase llamada **Estudiante** con tres atributos:

`nombre` (`String`), `apellido` (`String`) y `promedio_calificación` (`double`)

Crea una segunda clase llamada **Ejercicio32** con un método `main`. Declara una referencia a un objeto **Estudiante** e imprime directamente esta variable en pantalla. ¿Qué sucede? Inicializa la referencia al objeto con un valor `null`. Imprime la variable y el atributo `nombre`. ¿Qué sucede? ¿Por qué? Agrega el código adecuado para inicializar la referencia al objeto (usa el operador ‘`new`’). Imprime la variable y el atributo `nombre`. ¿Qué sucede?

Ejercicio 33. Inicialización de atributos

Extiende el ejercicio anterior para imprimir todos los atributos del objeto en pantalla. ¿Qué sucede? Asigna valores a los atributos del objeto en el método `main` e imprímelos de nuevo. ¿Qué sucede? Modifica la implementación de la clase *Estudiante* para asignar valores predefinidos a los atributos: `nombre` y `apellido` deben ser **??**, `promedio_calificación` debe ser 0. Declara una segunda referencia a un objeto *Estudiante* e inicialízala con el operador ‘`new`’. Imprime los valores de los atributos de este segundo objeto en pantalla. ¿Qué sucede? ¿Cuál es la diferencia con respecto al caso anterior? Asigne otros valores a los atributos del segundo objeto. ¿Es posible?

Ejercicio 34. Asignación directa de objetos

Amplíe el ejercicio anterior asignando el primer objeto *Estudiante* al segundo objeto *Estudiante*. Imprima en pantalla los valores de los atributos de ambos objetos. Asigne el valor **John**

al atributo *nombre* del segundo objeto. Imprima en pantalla el valor del atributo *nombre* del primer objeto. ¿Qué sucede?

Ejercicio 35. Depuración de programas con objetos

Ejecute el programa desarrollado en los ejercicios 33 y 34 en modo de depuración: utilice un punto de interrupción y la ejecución paso a paso. ¿Puede observar cómo cambian los valores de los objetos y sus atributos? ¿Cuáles son las ventajas del depurador en este caso?