

# Trabajo Práctico № 2

## Paradigmas de Programación

Dr. Pablo Javier Vidal Unidad 2

### Ejercicio 1.


1. Indique cuales son los valores de las variables al finalizar cada uno de los siguientes segmentos de código JAVA, asumiendo que todas han sido declaradas como enteras

```
1 // Ejercicio 1.1
2 public static void ejercicio1(){
3     int a; int b; int c;
4
5     a = 2; a++;
6     System.out.println("a. " + a); //a=3
7     a = 2; b = a++;
8     System.out.println("b. " + a + " " + b); //a=3, b=2
9     a = 2; b = ++a;
10    System.out.println("c. " + a + " " + b); //a=3, b=3
11    a = 2; b = 1; b += a;
12    System.out.println("d. " + a + " " + b); //a=2, b=3
13    b = 1; b *= 5;
14    System.out.println("e. " + b); //b=5
15    //a = 2; b = 1; b += -a + 5;
16    System.out.println("f. ERROR"); //error
17    //a = 2; b = 2; a += b-;
18    System.out.println("g. ERROR"); //error
19    a = 2; b = 3; c = 5; a++; b+= a; c *= b; b -= 3; a %= 2; c /= 5;
20    System.out.println("h. " + a + " " + b + " " + c); //a=1, b=3, c=6
21    a = 1; b = 2; b++; b = ++a; a*=2; b +=a; a = ++b + 2;
22    System.out.println("i. " + a + " " + b); //a=9, b=7
23    //a = 1; b = 4; a++; b += a; a *= 4; b -= a; ++b; a = ++b; b = -a + b;
24    System.out.println("j. ERROR"); //error
25 }
```

2. Reescriba las porciones de código del inciso (i) utilizando solamente los operadores =, +, -, \*, / y

```
1 // Ejercicio 1.2
2 public static void ejercicio2(){
3     int a; int b;
4
5     a=1; b=2; b=b+1; a=a+1; a=a*2; b=b+a; a=b+2;
6     System.out.println("i. " + a + " " + b); //a=9, b=7
7 }
```

3. Indique cuales son los valores de las variables al finalizar cada uno de los siguientes segmentos de código Java, asumiendo que a y b se declararon como enteras y v de tipo boolean.



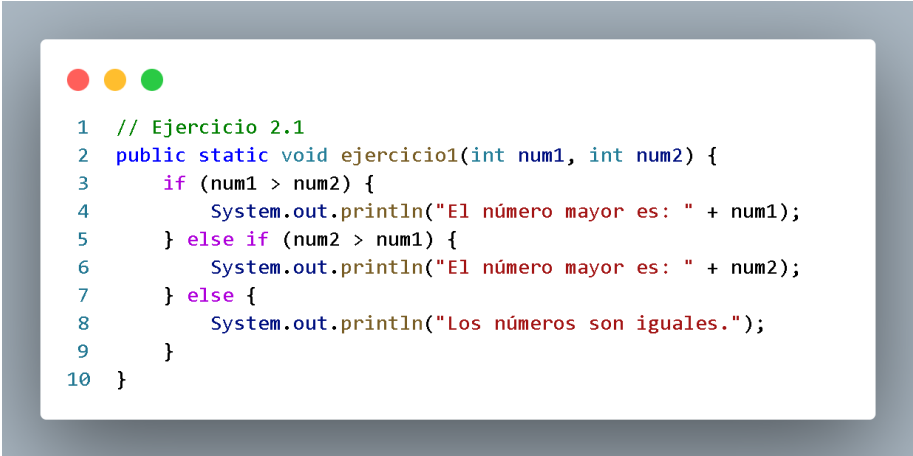
```
1 // Ejercicio 1.3
2 public static void ejercicio3(){
3     int a; int b; boolean v;
4
5     a = 1; b = 2; v = (a++ < b);
6     System.out.println("a. " + v); //true
7     a = 1; b = 2; v = (++a < b);
8     System.out.println("b. " + v); //false
9     a = 1; b = 2; v = (++a >= b);
10    System.out.println("c. " + v); //true
11    // a = 1; v = (a);
12    System.out.println("d. ERROR"); //error
13    a = 1; v = (a != 1);
14    System.out.println("e. " + v); //false
15    a = 1; b = 2; v=(b++ < 10 && a == 1);
16    System.out.println("f. " + v); //true
17    a = 1; b = 2; v=(b == 1 && a >= 1);
18    System.out.println("g. " + v); //false
19    a = 1; b = 2; v=(b < 10 || a++ == 2);
20    System.out.println("h. " + v); //true
21    // a = 1; b = 2; v = (-b <= 1 || a <= 10);
22    System.out.println("i. ERROR"); //error
23    // a = 1; b = 2; v = (a- == 2 || b == 1);
24    System.out.println("j. ERROR"); //error
25    a = 1; b = 2; v =! (a == 1 && ++b == 1);
26    System.out.println("k. " + v); //true
27 }
```

# Introducción a Java

Para la realización de los siguientes ejercicios, el alumno deberá definir una clase y por cada ítem solicitado deberá colocar la implementación del mismo en un método

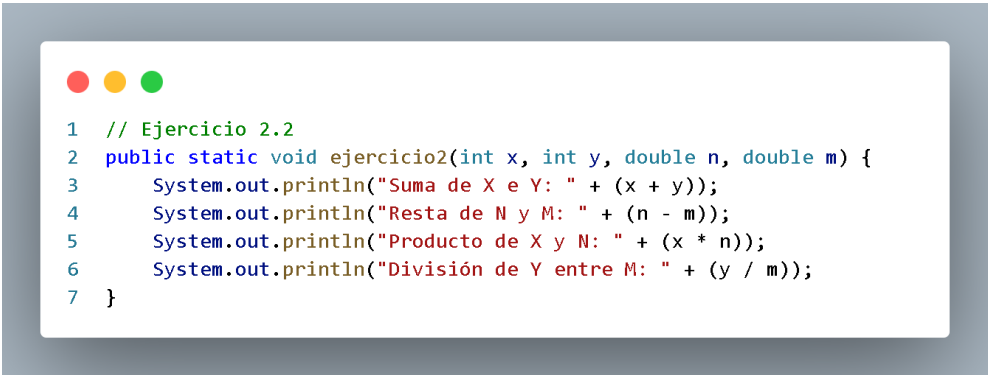
## Ejercicio 2.

1. Implementar un programa que dado dos números informe cual es el mayor.



```
1 // Ejercicio 2.1
2 public static void ejercicio1(int num1, int num2) {
3     if (num1 > num2) {
4         System.out.println("El número mayor es: " + num1);
5     } else if (num2 > num1) {
6         System.out.println("El número mayor es: " + num2);
7     } else {
8         System.out.println("Los números son iguales.");
9     }
10 }
```

2. Declarar dos variables X e Y de tipo int, dos variables N y M de tipo *double* y asignar a cada una un valor. A continuación mostrar por pantalla el resultado de una serie de operaciones matemáticas básicas entre ellas( las operaciones son a elección del alumno).



```
1 // Ejercicio 2.2
2 public static void ejercicio2(int x, int y, double n, double m) {
3     System.out.println("Suma de X e Y: " + (x + y));
4     System.out.println("Resta de N y M: " + (n - m));
5     System.out.println("Producto de X y N: " + (x * n));
6     System.out.println("División de Y entre M: " + (y / m));
7 }
```

3. Declarar cuatro variables enteras A, B, C y D y asignarle un valor diferente a cada una. A continuación, realizar las instrucciones necesarias para que: B tome el valor de C, C tome el valor de A, A tome el valor de D, D tome el valor de B. Mostrar los valores iniciales y los valores finales de cada variable.

```
1 // Ejercicio 2.3
2 public static void ejercicio3(int a, int b, int c, int d) {
3     System.out.println("Valores iniciales: A=" + a + ", B=" + b + ", C=" + c + ", D=" + d);
4     int temp = b;
5     b = c;
6     c = a;
7     a = d;
8     d = temp;
9     System.out.println("Valores finales: A=" + a + ", B=" + b + ", C=" + c + ", D=" + d);
10 }
```

4. Calcular el volumen de una esfera. Recordar que para calcular el volumen se debe utilizar la siguiente fórmula  $V = \frac{4}{3} * \pi * r^3$  siendo  $r^3$  el radio de la esfera.

```
1 // Ejercicio 2.4
2 public static void ejercicio4(double radio) {
3     double volumen = (4.0 / 3.0) * Math.PI * Math.pow(radio, 3);
4     System.out.println("El volumen de la esfera es: " + volumen);
5 }
```

5. Calcular el área de un triángulo a partir de la longitud de sus lados. El cálculo de un semi perímetro es  $p = \frac{1}{2} * (a + b + c)$  siendo a, b y c las longitudes.

```
1 // Ejercicio 2.5
2 public static void ejercicio5(double a, double b, double c) {
3     double semiPerimetro = 0.5 * (a + b + c);
4     double area = Math.sqrt(semiPerimetro * (semiPerimetro - a) * (semiPerimetro - b) * (semiPerimetro - c));
5     System.out.println("El área del triángulo es: " + area);
6 }
```

## Ejercicio 3.

1. Implementar un programa que defina un vector de tamaño 10 y completar con números del 1 al 10.

```
1 // Ejercicio 3.1
2 public static void ejercicio1() {
3     int[] vector = new int[10];
4     for (int i = 0; i < 10; i++) {
5         vector[i] = i + 1;
6     }
7     System.out.println(Arrays.toString(vector));
8 }
```

2. Reutilizando el código anterior, definir un programa que muestre la tabla del 3, luego la del 4 y finalmente la del 8.

```
1 // Ejercicio 3.2
2 public static void ejercicio2(int numero) {
3     System.out.println("Tabla del " + numero + ":");
4     for (int i = 1; i <= 10; i++) {
5         int resultado = numero * i;
6         System.out.println(numero + " x " + i + " = " + resultado);
7     }
8 }
```

3. Definir un vector y completar cada posición con números del 1 al 20. Una vez completado el vector sumar los valores generados. NO se debe realizar la misma operación de asignación del dato y suma en la misma estructura de repetición. NO utilizar una estructura for.

```
1 // Ejercicio 3.3
2 public static void ejercicio3() {
3     int[] vector = new int[20];
4
5     int index = 0;
6     int numero = 1;
7     while (index < vector.length) {
8         vector[index] = numero;
9         index++;
10        numero++;
11    }
12
13    int suma = 0;
14    index = 0;
15    while (index < vector.length) {
16        suma += vector[index];
17        index++;
18    }
19    System.out.println(Arrays.toString(vector));
20    System.out.println("Vector completado y suma de valores: " + suma);
21 }
```

4. Definir una matriz de 2×2. Para cada posición (i,j), asignar el valor resultante de la operación  $valor = (2 * i) + (j + 3)$ .

```
1 // Ejercicio 3.4
2 public static void ejercicio4() {
3     int[][] matriz = new int[2][2];
4     for (int i = 0; i < 2; i++) {
5         for (int j = 0; j < 2; j++) {
6             matriz[i][j] = (2 * i) + (j + 3);
7         }
8     }
9     System.out.println(Arrays.deepToString(matriz));
10 }
```

5. Definir una matriz de 20×20 elementos. Completar dicha matriz con los valores generados por  $valor = fila * 20 + columna$ .

```
1 // Ejercicio 3.5
2 public static void ejercicio5() {
3     int[][] matriz = new int[20][20];
4     for (int i = 0; i < 20; i++) {
5         for (int j = 0; j < 20; j++) {
6             matriz[i][j] = i * 20 + j;
7         }
8     }
9     System.out.println(Arrays.deepToString(matriz));
10 }
```

## Ejercicio 4.

1. Diseñar un programa que permita ingresar diferentes caracteres, el programa solo se detendrá si se ingresa una letra X.

```
1 // Ejercicio 4.1
2 public void ejercicio1() {
3     char caracter;
4     do {
5         System.out.print("Ingrese un caracter ('X' para detener): ");
6         caracter = scanner.next().charAt(0);
7     } while (caracter != 'X');
8 }
```

2. Leer un nombre y muestre por pantalla: "Buenos días nombre\_introducido".

```
1 // Ejercicio 4.2
2 public void ejercicio2() {
3     scanner.nextLine(); // Consumir el salto de línea pendiente
4     System.out.print("Ingrese su nombre: ");
5     String nombre = scanner.nextLine();
6     System.out.println("Buenos días " + nombre);
7 }
```

3. Leer un número entero y calcular si es par o impar.

```
1 // Ejercicio 4.3
2 public void ejercicio3() {
3     System.out.print("Ingrese un número entero: ");
4     int numero = scanner.nextInt();
5     if (numero % 2 == 0) {
6         System.out.println("El número es par.");
7     } else {
8         System.out.println("El número es impar.");
9     }
10 }
```

4. Leer un número por teclado que pida el precio de un producto (puede tener decimales) y calcule el precio final con IVA. El IVA será una constante que será del 21 %.

```
1 // Ejercicio 4.4
2 public void ejercicio4() {
3     final double IVA = 0.21;
4     System.out.print("Ingrese el precio del producto: ");
5     double precio = scanner.nextDouble();
6     double precioFinal = precio * (1 + IVA);
7     System.out.println("El precio final con IVA es: " + precioFinal);
8 }
```

5. Leer dos números por teclado y mostrar el resultado de la división del primero por el segundo. Se debe comprobar que el divisor no puede ser cero.

```
1 // Ejercicio 4.5
2 public void ejercicio5() {
3     System.out.print("Ingrese el primer número: ");
4     double numero1 = scanner.nextDouble();
5     System.out.print("Ingrese el segundo número: ");
6     double numero2 = scanner.nextDouble();
7
8     if (numero2 != 0) {
9         double resultado = numero1 / numero2;
10        System.out.println("El resultado de la división es: " + resultado);
11    } else {
12        System.out.println("Error: No se puede dividir entre cero.");
13    }
14 }
```

6. Calcular el promedio de una serie de números que se leen por teclado.

```
1 // Ejercicio 4.6
2 public void ejercicio6() {
3     System.out.print("Ingrese la cantidad de números a promediar: ");
4     int cantidad = scanner.nextInt();
5     double suma = 0;
6
7     for (int i = 1; i <= cantidad; i++) {
8         System.out.print("Ingrese el número " + i + ": ");
9         double numero = scanner.nextDouble();
10        suma += numero;
11    }
12
13    double promedio = suma / cantidad;
14    System.out.println("El promedio de los números es: " + promedio);
15 }
```



7. Pedir un día de la semana y que nos diga si es un día laboral o no. Usar una instrucción switch para ello.

```
1 // Ejercicio 4.7
2 public void ejercicio7() {
3     System.out.print("Ingrese un día de la semana (1:lunes,2:martes,3:miercoles,4:jueves,5:viernes,6:sabado,7:domingo): ");
4     int dia = scanner.nextInt();
5
6     switch (dia) {
7         case 1: case 2: case 3: case 4: case 5:
8             System.out.println("Es un día laboral.");
9             break;
10        case 6: case 7:
11            System.out.println("No es un día laboral.");
12            break;
13        default:
14            System.out.println("Día inválido.");
15        }
16    }
```

## Estructuras

### Ejercicio 5.

1. Analice si los siguientes segmentos de instrucciones son equivalentes en términos del ámbito y los valores de las variables

```
1
2 float sum=0; int i;
3 for (i = 10; 1/i > sum; i--){
4     sum = sum + 1/i;
5 }
```

```
1 float sum=0;
2 for (int i = 10; 1/i > sum; i--){
3     sum = sum + 1/i;
4 }
5
```

```
1 for (int sum=0, i = 10;
2     1/i > sum; i--){
3     sum = sum + 1/i;
4 }
5
```

Los tres segmentos de instrucciones son equivalentes en términos del ámbito y los valores de las variables. Cada uno de ellos realiza una operación similar que calcula la suma de los valores  $1/i$  para  $i$  empezando en 10 hasta que  $1/i$  sea mayor que la suma.

- float sum: En todos los ejercicios, se declara y se inicializa la variable sum con 0. Esta variable se utiliza para llevar un seguimiento de la suma acumulativa de los valores  $1/i$ .
- int i: En los ejercicios 1a y 1b, la variable i se declara fuera del bucle, mientras que en el ejercicio 1c, se declara dentro del bucle, pero esto no afecta el resultado ya que i no se usa fuera del bucle.

**2. Considere los siguientes fragmentos extraídos de un programa. Asuma la siguiente declaración de variables:**

```
int a,b;  
char c;
```

```
a= 1;  
b= 1;  
if((a=2)>(b=1))  
    a = b;  
else  
    b = a;
```

```
a= 1;  
b= 1;  
if (a > b)  
    b = a;  
    a = 0;  
else  
    a = b;  
    b = 0;
```

```
a = 1;  
b = 1;  
if (a == b) {  
    a=1;  
    b=2;  
}  
else {  
    a=2;  
    b=1;  
}
```

Los tres segmentos de instrucciones (ejercicios 5.2a, 5.2b y 5.2c) son diferentes en términos de sus operaciones y, por lo tanto, generan resultados diferentes. Aquí está el análisis de cada uno de ellos:

**Ejercicio 5.2a:**

- Se inicializan las variables a y b con 1.
- Se evalúa la condición  $(a=2) > (b=1)$ , lo que asigna el valor 2 a a y el valor 1 a b, y luego verifica si 2 es mayor que 1. Esto es verdadero.
- En el bloque if, a se establece en el valor de b, que es 1.
- Luego, se imprime "a: 1, b: 1".

**Ejercicio 5.2b:**

- Se inicializan las variables a y b con 1.
- Se evalúa la condición  $a > b$ , lo que es verdadero.
- En el bloque if, b se establece en el valor de a, que es 1, y a se establece en 0.
- Luego, se imprime "a: 0, b: 1".

**Ejercicio 5.2c:**

- Se inicializan las variables a y b con 1.
- Se evalúa la condición  $a == b$ , lo que es verdadero.
- En el bloque if, a se establece en 1, y b se establece en 2.
- Luego, se imprime "a: 1, b: 2".

## Ejercicio 6.

1. Implementar un programa donde se tiene el dato del día (puede ser número o letra) e informar a qué día corresponde.

```
1 // Ejercicio 6.1
2 public void ejercicio1() {
3     System.out.print("Ingresa un caracter(1:L, 2:M, 3:X, 4:J, 5:V, 6:S, 7:D ):");
4     char dia = scanner.next().charAt(0);
5     switch (dia) {
6         case 'L': case '1':
7             System.out.println("Lunes");
8             break;
9         case 'M': case '2':
10            System.out.println("Martes");
11            break;
12         case 'X': case '3':
13            System.out.println("Miércoles");
14            break;
15         case 'J': case '4':
16            System.out.println("Jueves");
17            break;
18         case 'V': case '5':
19            System.out.println("Viernes");
20            break;
21         case 'S': case '6':
22            System.out.println("Sábado");
23            break;
24         case 'D': case '7':
25            System.out.println("Domingo");
26            break;
27         default:
28            System.out.println("Dia inválido");
29     }
30 }
```

2. Implementar un programa que sume los números del 1 al 10. Utilizar las tres estructuras de repetición: for, do-while, while.

```
1 // Ejercicio 6.2
2 public void ejercicio2() {
3     int sumaFor = 0;
4     for (int i = 1; i <= 10; i++) {
5         sumaFor += i;
6     }
7     System.out.println("Suma usando for: " + sumaFor);
8
9     int sumaWhile = 0;
10    int i = 1;
11    while (i <= 10) {
12        sumaWhile += i;
13        i++;
14    }
15    System.out.println("Suma usando while: " + sumaWhile);
16
17    int sumaDoWhile = 0;
18    int j = 1;
19    do {
20        sumaDoWhile += j;
21        j++;
22    } while (j <= 10);
23    System.out.println("Suma usando do-while: " + sumaDoWhile);
24 }
```

### 3. Dado un valor booleano informar si es verdadero o falso.

```
1 // Ejercicio 6.3
2 public void ejercicio3() {
3     System.out.print("Ingrese true o false: ");
4     String input = scanner.next();
5     boolean valor = Boolean.parseBoolean(input);
6     if (valor) {
7         System.out.println("Es verdadero");
8     } else {
9         System.out.println("Es falso");
10    }
11 }
```

### 4. Dado un carácter numérico informar si es un dígito o no.

```
1 // Ejercicio 6.4
2 public void ejercicio4() {
3     System.out.print("Ingrese un caracter para identificar si es un digito o no: ");
4     char caracter = scanner.next().charAt(0);
5     if (Character.isDigit(caracter)) {
6         System.out.println("Es un dígito");
7     } else {
8         System.out.println("No es un dígito");
9     }
10 }
```

## Ejercicio 7.

¿Qué afirmación es cierta? Justifique su respuesta teniendo en cuenta el siguiente código:

```
1 Ordenador escritorio; Ordenador portatil;
2 escritorio = new Ordenador();
3 escritorio.precio(900);
4 portatil = new Ordenador();
5 portatil.precio(1100);
6 portatil = escritorio;
7 escritorio = null;
```

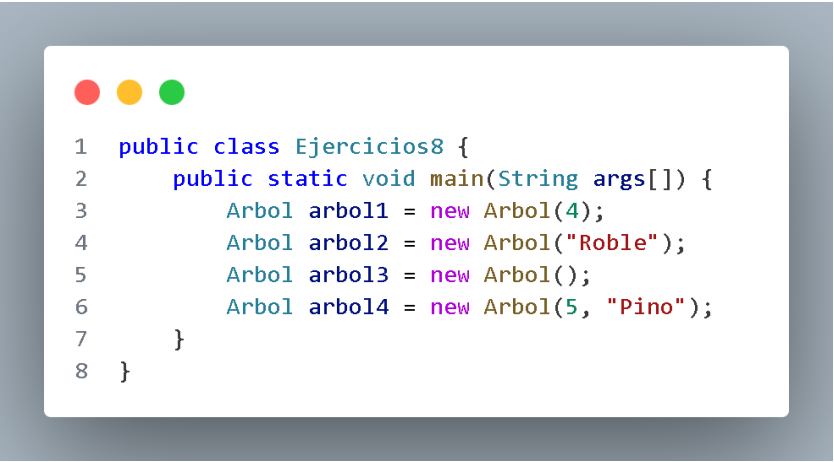
1. Cuando se ejecuta la línea 5, la instancia escritorio cuesta 1100. // NO cierta
2. Cuando se ejecuta la línea 5, la instancia portátil cuesta 1100. // cierta
3. Al final tanto el objeto escritorio como el objeto portátil apuntan a null. // NO cierta
4. Al final sólo queda un objeto de tipo Ordenador con precio 1100. // NO cierta
5. Al final sólo queda un objeto de tipo Ordenador con precio 900. // cierta
6. Al final hay dos objetos de tipo Ordenador, uno con precio 900 y otro con precio 1100. // NO cierta

## Ejercicio 8.

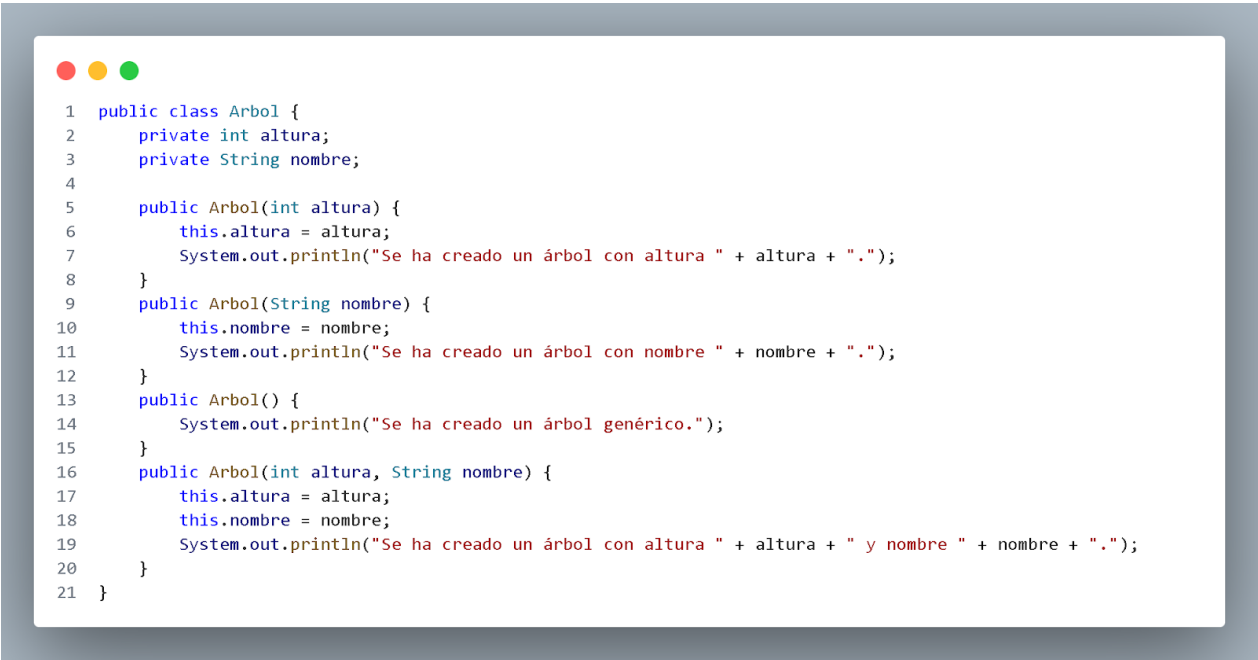
Crear los constructores necesarios para poder instanciar un Arbol con su altura y nombre de la siguiente manera:

```
1 public static void main(String args[]) {
2     Arbol arbol1 = new Arbol(4);
3     Arbol arbol2 = new Arbol("Roble");
4     Arbol arbol3 = new Arbol();
5     Arbol arbol4 = new Arbol(5, "Pino");
6 }
```

Cada constructor deberá mostrar un mensaje por consola con información, si no tiene datos es un árbol genérico.



```
1 public class Ejercicios8 {
2     public static void main(String args[]) {
3         Arbol arbol1 = new Arbol(4);
4         Arbol arbol2 = new Arbol("Roble");
5         Arbol arbol3 = new Arbol();
6         Arbol arbol4 = new Arbol(5, "Pino");
7     }
8 }
```



```
1 public class Arbol {
2     private int altura;
3     private String nombre;
4
5     public Arbol(int altura) {
6         this.altura = altura;
7         System.out.println("Se ha creado un árbol con altura " + altura + ".");
8     }
9     public Arbol(String nombre) {
10        this.nombre = nombre;
11        System.out.println("Se ha creado un árbol con nombre " + nombre + ".");
12    }
13    public Arbol() {
14        System.out.println("Se ha creado un árbol genérico.");
15    }
16    public Arbol(int altura, String nombre) {
17        this.altura = altura;
18        this.nombre = nombre;
19        System.out.println("Se ha creado un árbol con altura " + altura + " y nombre " + nombre + ".");
20    }
21 }
```

## Clases con Java

Ejercicio 9. Para cada una de las clases, definir siempre los métodos getters y setters aparte de los solicitados.

1. Definir la clase **Persona**, reutilizando lo definido hasta el momento. Implementar al menos 4 métodos propios del comportamiento de la clase **Persona** y además los métodos **getters** y **setters**. Finalmente, generar una instancia en el método **main**.

```
1 public class Persona {
2     private String nombre;
3     private int edad;
4     private String genero;
5     private String ocupacion;
6
7     // Constructor
8     public Persona(String nombre, int edad, String genero, String ocupacion) {
9         this.nombre = nombre;
10        this.edad = edad;
11        this.genero = genero;
12        this.ocupacion = ocupacion;
13    }
```

```
1 // Métodos propios
2 public void saludar() {
3     System.out.println("Hola, soy " + nombre + ".");
4 }
5
6 public void cumplirAnios() {
7     edad++;
8     System.out.println("¡Feliz cumpleaños! Ahora tengo " + edad + " años.");
9 }
10
11 public void trabajar() {
12     System.out.println("Estoy trabajando como " + ocupacion + ".");
13 }
14
15 public void descansar() {
16     System.out.println("Voy a descansar un rato.");
17 }
```

```
1 // Getters y setters
2 public String getNombre() {
3     return nombre;
4 }
5 public void setNombre(String nombre) {
6     this.nombre = nombre;
7 }
8 public int getEdad() {
9     return edad;
10 }
11 public void setEdad(int edad) {
12     this.edad = edad;
13 }
14 public String getGenero() {
15     return genero;
16 }
17 public void setGenero(String genero) {
18     this.genero = genero;
19 }
20 public String getOcupacion() {
21     return ocupacion;
22 }
23 public void setOcupacion(String ocupacion) {
24     this.ocupacion = ocupacion;
25 }
```

**2. Definir una clase Punto. Generar dos instancias P1 y P2. Comprobar la distancia que existe desde P1 a P2.**

```
1 public class Punto {
2     private double x;
3     private double y;
4
5     // Constructor
6     public Punto(double x, double y){
7         this.x = x;
8         this.y = y;
9     }
```


```
1 // Getters y setters
2 public double getX(){
3     return x;
4 }
5 public void setX(double x){
6     this.x = x;
7 }
8 public double getY(){
9     return y;
10 }
11 public void setY(double y){
12     this.y = y;
13 }
```

```
1 // Métodos propios
2 public double calcularDistancia(Punto otroPunto) {
3     double distanciaX = this.x - otroPunto.x;
4     double distanciaY = this.y - otroPunto.y;
5     return Math.sqrt(distanciaX * distanciaX + distanciaY * distanciaY);
6 }
7
```

**3. Diseñar una clase Sumatoria que tenga: una variable vector de tipo entero y dos métodos, uno que permita mostrar la tabla de multiplicar para cada variable y otro que permita devolver la suma. Debe tener un constructor por defecto que les asigne valores entre 1 y 10 o bien un constructor para ingresar los dos valores. En el main de prueba deberá generar dos instancias de la clase Sumatoria, mostrar las tablas de multiplicar y finalmente mostrar la suma de ambas variables por pantalla.**

```
1 public class Sumatoria {
2
3     private int[] vector;
4
5     // Constructor por defecto
6     public Sumatoria(){
7         vector = new int[2];
8         // Asignar valores aleatorios entre 1 y 10
9         for (int i = 0; i < vector.length; i++) {
10             vector[i] = (int) (Math.random() * 10) + 1;
11         }
12     }
```

```
1 // Constructor para ingresar valores
2 public Sumatoria(int valor1,int valor2){
3     vector = new int[2];
4     vector[0] = valor1;
5     vector[1] = valor2;
6 }
```




```

1 // Método para mostrar la tabla de multiplicar
2 public void mostrarTablaDeMultiplicar() {
3     for (int i = 1; i <= 10; i++){
4         System.out.println(vector[0] + " X " + i + " = " + vector[0]*i);
5         System.out.println(vector[1] + " X " + i + " = " + vector[1]*i);
6     }
7 }
8
9 // Método para calcular la suma
10 public void calcularSuma(){
11     System.out.println(vector[0] + " + " + vector[1] + " = " + vector[0]+vector[1]);
12 }

```


4. Crear una clase *Libro* que contenga los siguientes atributos: ISBN, Título, Autor, Número de páginas. Definir 5 instancias y para cada una de ellas informar mediante un método el número de ISBN, el título y el autor.



```

1 public class Libro {
2     private String isbn;
3     private String titulo;
4     private String autor;
5     private int numPaginas;
6
7     // Constructor
8     public Libro(String isbn, String titulo, String autor, int numPaginas){
9         this.isbn = isbn;
10        this.titulo = titulo;
11        this.autor = autor;
12        this.numPaginas = numPaginas;
13    }

```



```

1 // Método para obtener la información de ISBN, Título y Autor
2 public void obtenerInformacion() {
3     System.out.println("ISBN: " + isbn);
4     System.out.println("Título: " + titulo);
5     System.out.println("Autor: " + autor);
6 }

```



**5. Crear una clase *Fraccion* con métodos para sumar, restar, multiplicar y dividir fracciones. En cada método se debe mostrar el resultado de dicha operación.**

```
1 public class Fraccion {
2     private int numerador;
3     private int denominador;
4
5     // Constructor
6     public Fraccion(int numerador, int denominador){
7         this.numerador = numerador;
8         if (denominador != 0) {
9             this.denominador = denominador;
10        } else {
11            throw new IllegalArgumentException("El denominador no puede ser cero.");
12        }
13    }
```

```
1 // Método para sumar dos fracciones
2 public void sumar(Fraccion otraFraccion) {
3     int nuevoDenominador = this.denominador * otraFraccion.denominador;
4     int nuevoNumerador = this.numerador * otraFraccion.denominador + otraFraccion.numerador * this.denominador;
5     System.out.println("La suma es: " + nuevoNumerador + "/" + nuevoDenominador);
6 }
7
8 // Método para restar dos fracciones
9 public void restar(Fraccion otraFraccion) {
10    int nuevoDenominador = this.denominador * otraFraccion.denominador;
11    int nuevoNumerador = this.numerador * otraFraccion.denominador - otraFraccion.numerador * this.denominador;
12    System.out.println("La resta es: " + nuevoNumerador + "/" + nuevoDenominador);
13 }
14
15 // Método para multiplicar dos fracciones
16 public void multiplicar(Fraccion otraFraccion) {
17     int nuevoNumerador = this.numerador * otraFraccion.numerador;
18     int nuevoDenominador = this.denominador * otraFraccion.denominador;
19     System.out.println("La multiplicación es: " + nuevoNumerador + "/" + nuevoDenominador);
20 }
21
22 // Método para dividir dos fracciones
23 public void dividir(Fraccion otraFraccion) {
24     if (otraFraccion.numerador != 0) {
25         int nuevoNumerador = this.numerador * otraFraccion.denominador;
26         int nuevoDenominador = this.denominador * otraFraccion.numerador;
27         System.out.println("La división es: " + nuevoNumerador + "/" + nuevoDenominador);
28     } else {
29         throw new IllegalArgumentException("No se puede dividir por cero.");
30     }
31 }
```