

Pasaje de parámetros en Java

En Java los parámetros se pasan por valor. Pasaje por valor significa que cuando se invoca a un método, se pasan como argumentos al método una copia de cada parámetro actual.

```
package ayed.tp02;

public class PasajePorValor {
    public static int mult(int x, int y) {
        return x * y;
    }
    public static void main(String[] args) {
        int alto = 10;
        int ancho = 5;
        int area = mult(alto, ancho);
    }
}
```

Parámetros formales

Son los parámetros en la definición del método

Parámetros actuales/reales

Son los parámetros en la invocación al método

Dentro del cuerpo de un método se puede cambiar el valor de la copia que se recibe como parámetro, pero no tendrá efecto en el parámetro actual.

Pasaje de parámetros

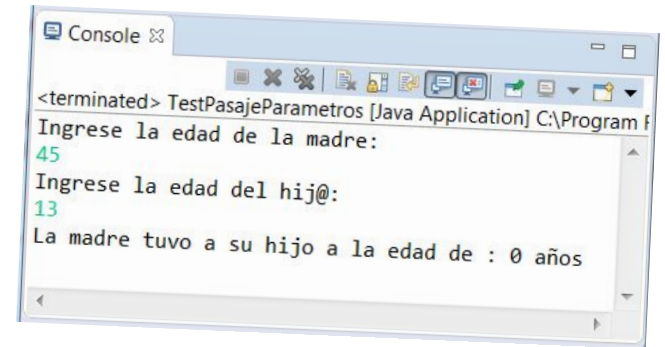
Tipo de dato primitivo

Cuando Java llama a un método, hace una copia de sus parámetros reales y envía las copias al método. Cuando el método finaliza, esas copias se descartan y los valores de las variables en el código principal son los mismos que antes.

```
package tp02.pasajeparametros;
import java.util.Scanner;

public class TestPasajeParametros {
    public static void pedirEdades(int edad1, int edad2) {
        Scanner input = new Scanner(System.in); // Create a Scanner object
        System.out.println("Ingrese la edad de la madre:");
        edad1 = input.nextInt();
        System.out.println("Ingrese la edad del hijo:");
        edad2 = input.nextInt();
    }

    public static void main(String[] args) {
        int edadMadre = 0;
        int edadHijo = 0;
        TestPasajeParametros.pedirEdades(edadMadre, edadHijo);
        System.out.println("La madre tuvo a su hijo a los: " + (edadMadre - edadHijo) + " años");
    }
}
```



Pasaje de parámetros

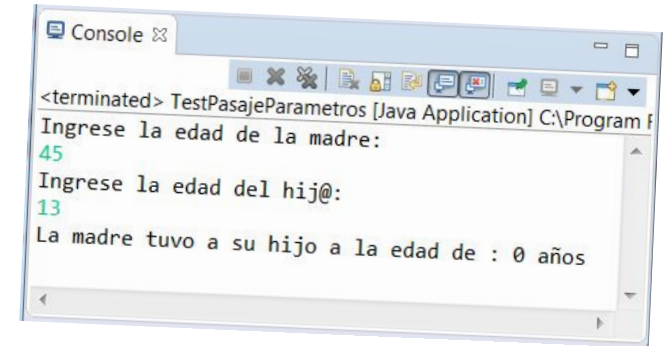
Clases wrapper y clase String

Las clases wrapper (**Integer**, **Character**, **Double** ..) y las instancias de **String** son *inmutables*., esto significa que su valor no puede cambiar, de manera que cuando se modifica el valor a una variable de estos tipos, se crea una nueva instancia con el nuevo valor y se le asigna a la variable.

```
package tp02.pasajeparametros;
import java.util.Scanner;

public class TestPasajeParametros {
    public static void pedirEdades(Integer edad1, Integer edad2) {
        Scanner input = new Scanner(System.in); // Create a Scanner object
        System.out.println("Ingrese la edad de la madre:");
        edad1 = input.nextInt();
        System.out.println("Ingrese la edad del hijo:");
        edad2 = input.nextInt();
    }

    public static void main(String[] args) {
        Integer edadMadre = 0;
        Integer edadHijo = 0;
        TestPasajeParametros.pedirEdades(edadMadre, edadHijo);
        System.out.println("La madre tuvo a su hijo a los: " + (edadMadre - edadHijo) + " años");
    }
}
```



Pasaje de parámetros

Tipo de dato referencial

En Java, podemos pasar como parámetro una referencia a un objeto y con ella cambiar el estado (valor de sus variables) del objeto, pero no la referencia.

```
public class PasajePorValor {  
    public static void cambiarNombre(Contacto c) {  
        c.setNombre("Pilar");  
    }  
  
    public static void main(String[] args){  
        Contacto c = new Contacto();  
        c.setNombre("Lucia");  
        PasajePorValor.cambiarNombre(c);  
        System.out.println(c.getNombre());  
    }  
}
```

¿Qué sucedería si reemplazamos esta línea por estas dos?

```
c=new Contacto();  
c.setNombre("Juan");
```

Se pasa una copia de la referencia a c

Calcular el máximo de un arreglo

Dado un arreglo con valores de tipo int, se desea calcular el máximo valor del arreglo.

- ¿Cómo implementamos el método `maximo(arreglo)`?
- ¿Es un método de instancia o de clase?
- ¿Qué parámetros le pasamos?

```
package tp02.ejercicio5;

public class Test {

    public static void main(String[] args) {
        int[] datos = {3, 4, 5, 8, 0};
        int max = Calculadora.maximo(datos);
        System.out.println("El máximo es " + max);
    }
}
```

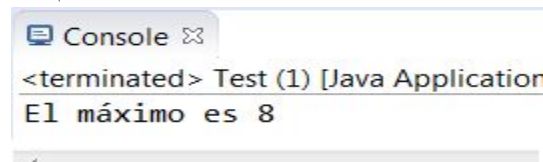
Calcular el máximo de un arreglo

La manera más simple es usar la sentencia **return** para devolver el máximo del arreglo.

```
package ayed.tp02;
public class Calculadora {
    public static int maximo(int[] datos) {
        int max = 0;
        for (int i = 0; i < datos.length; i++) {
            if (datos[i] > max)
                max = datos[i];
        }
        return max;
    }
}
```

```
package ayed.tp02;

public class Test {
    public static void main(String[] args) {
        int[] datos = {3, 4, 5, 8, 0};
        int max = Calculadora.maximo(datos);
        System.out.println("El máximo es " + max);
    }
}
```



Console

<terminated> Test (1) [Java Application]

El máximo es 8

¿Cómo podríamos hacer si tenemos que devolver el máximo y el mínimo?

Calcular el máximo y el mínimo de un arreglo

De la misma manera que devolvemos un int, podemos devolver un objeto con el máximo y el mínimo.

```
package ayed.tp02;

public class Calculadora {
    . . .

    public static Datos maxmin(int[] datos) {
        int max = 0, min = 0;
        for (int i = 0; i < datos.length; i++) {
            if (datos[i] > max) max = datos[i];
            if (datos[i] < min) min = datos[i];
        }
        Datos obj = new Datos();
        obj.setMax(max);
        obj.setMin(min);
        return obj;
    }
}
```

```
package ayed.tp02;

public class Datos {
    private int min;
    private int max;
    public int getMin() {
        return min;
    }
    public void setMin(int min) {
        this.min = min;
    }
    public int getMax() {
        return max;
    }
    public void setMax(int max) {
        this.max = max;
    }
}
```

En este caso creamos un objeto que mantiene el máximo y el mínimo y devolvemos ese objeto.

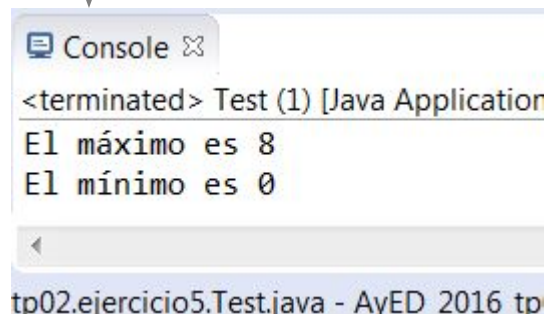
Calcular el mínimo y el máximo de un arreglo

Ahora al invocar al método nos devuelve un objeto:

```
package tp02.ejercicio5;

public class Test {

    public static void main(String[] args) {
        int[] datos = {3, 4, 5, 8, 0};
        Datos maxmin = Calculadora.maxmin(datos);
        System.out.println("El máximo es " + maxmin.getMax());
        System.out.println("El mínimo es " + maxmin.getMin());
    }
}
```



```
<terminated> Test (1) [Java Application]
El máximo es 8
El mínimo es 0

tp02.ejercicio5.Test.java - AyED 2016 tp
```