

JAVA



UTILIZACIÓN DE SUBPROGRAMAS Y CLASES DE USO COMÚN

1. Los métodos estáticos

1.- Los métodos estáticos

- Hasta ahora, todos los programas que hemos realizado, se han desarrollado íntegramente en el método main. Esto es válido para pequeños programas pero, cuando tenemos programas con cientos de líneas de código, lo más correcto es dividir el programa en pequeños subprogramas (métodos) que tengan una determinada funcionalidad.
- Para realizar nuestro primeros subprogramas, vamos a utilizar métodos estáticos, que son aquellos métodos que pertenecen a la clase y que no necesitan ser instanciados por un objeto.

1.- Los métodos estáticos

- Aunque en este tema utilicemos los métodos estáticos para aprender a utilizar subprogramas, en el tema siguiente veremos que suelen emplearse para realizar operaciones comunes a todos los objetos de la clase.
- Se cargan en memoria en tiempo de compilación y no a medida que se ejecutan las líneas de código del programa. Van precedidos del modificador static.
- Ejemplo:

```
public static void multiplicar (int numero1, int numero2){
```

5

PROGRAMACIÓN

1.- Los métodos estáticos

- Para invocar a un método estático no se necesita crear un objeto de la clase en la que se define:
 - Si se invoca desde la clase en la que se encuentra definido, basta con escribir su nombre.
 - Si se le invoca desde una clase distinta, debe anteponerse a su nombre, el de la clase en la que se encuentra seguido del operador punto (.) <NombreClase>.metodoEstatico

6

PROGRAMACIÓN

1.- Los métodos estáticos

- Vamos a ver un ejemplo en el que asignamos en el main una variable edad y otra variable altura, y estos 2 datos se los pasamos a un método para que los muestre:

```
package edadyaltura2;

/**
 * @author OLG
 */
public class EdadyAltura2 {

    public static void muestraEdadyAltura(byte e, float a) {
        System.out.println("Tengo " + e + " años");
        System.out.println("Mido " + a);
    }

    public static void main(String[] args) {
        byte edad;
        float altura;
        edad = 55;
        altura = 1.75F;
        muestraEdadyAltura(edad, altura);
    }
}
```

Input - EdadyAltura2 (run) ×

```
run:
Tengo 55 años
Mido 1.75
```

PROGRAMACIÓN

7

1.- Los métodos estáticos

- Otra forma de invocar al método estático es anteponiendo el nombre de la clase:

```
package edadyaltura2;

/**
 * @author OLG
 */
public class EdadyAltura2 {

    public static void muestraEdadyAltura(byte e, float a) {
        System.out.println("Tengo " + e + " años");
        System.out.println("Mido " + a);
    }

    public static void main(String[] args) {
        byte edad;
        float altura;
        edad = 55;
        altura = 1.75F;
        EdadyAltura2.muestraEdadyAltura(edad, altura);
    }
}
```

EdadyAltura2.EdadyAltura2 > main >

Input - EdadyAltura2 (run) ×

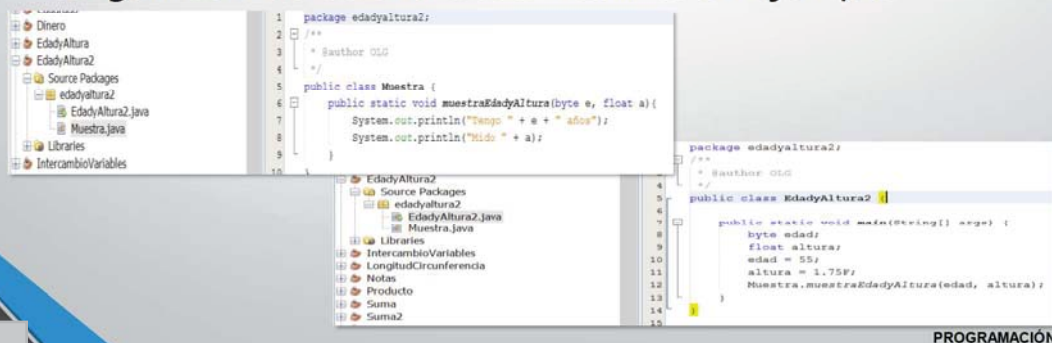
```
run:
Tengo 55 años
Mido 1.75
```

PROGRAMACIÓN

8

1.- Los métodos estáticos

- La forma de invocar al método estático anteponiendo el nombre de la clase es muy útil en aquellos proyectos que tengamos más de una clase. Veamos un ejemplo:



```
package edadyaltura2;
/**
 * @author OLG
 */
public class Muestra {
    public static void muestraEdadyAltura(byte e, float a){
        System.out.println("Tengo " + e + " años");
        System.out.println("Mido " + a);
    }
}

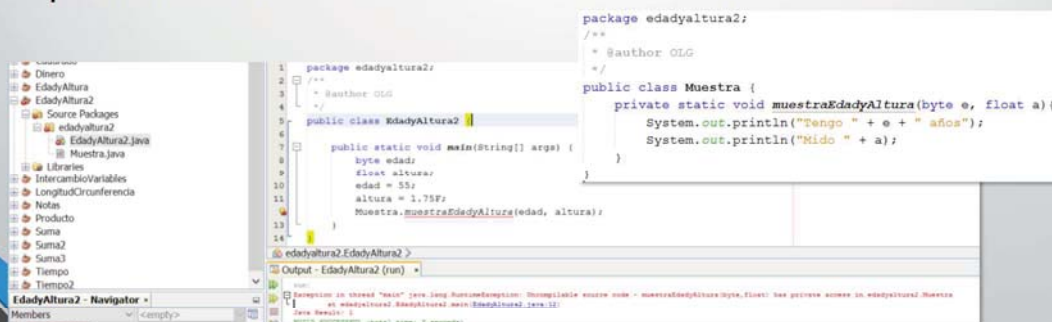
package edadyaltura2;
/**
 * @author OLG
 */
public class EdadyAltura2 {
    public static void main(String[] args) {
        byte edad;
        float altura;
        edad = 55;
        altura = 1.75f;
        Muestra.muestraEdadyAltura(edad, altura);
    }
}
```

PROGRAMACIÓN

9

1.- Los métodos estáticos

- Fíjate que el método `muestraEdadyAltura` es público ("public"), ya que si lo pusiéramos privado ("private") no podríamos acceder a él desde la otra clase:



```
package edadyaltura2;
/**
 * @author OLG
 */
public class Muestra {
    private static void muestraEdadyAltura(byte e, float a){
        System.out.println("Tengo " + e + " años");
        System.out.println("Mido " + a);
    }
}

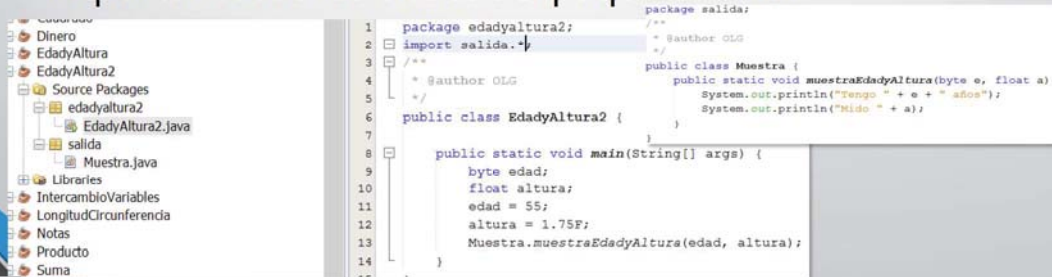
package edadyaltura2;
/**
 * @author OLG
 */
public class EdadyAltura2 {
    public static void main(String[] args) {
        byte edad;
        float altura;
        edad = 55;
        altura = 1.75f;
        Muestra.muestraEdadyAltura(edad, altura);
    }
}
```

PROGRAMACIÓN

10

1.- Los métodos estáticos

- Ahora, vamos a crear un nuevo paquete llamado salida y, dentro de él, creo la clase Muestra. Si quiero llamar a sus métodos desde la clase EdadyAltura2 tendré que importar todas las clases del paquete salida:



```

package edadyaltura2;
import salida.*;

/**
 * @author OLG
 */
public class EdadyAltura2 {

    public static void main(String[] args) {
        byte edad;
        float altura;
        edad = 55;
        altura = 1.75F;
        Muestra.muestraEdadyAltura(edad, altura);
    }
}
    
```

```

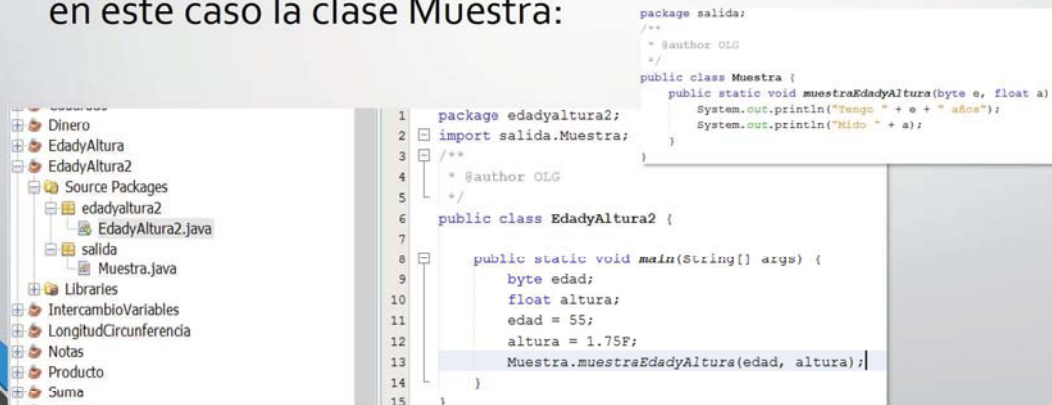
package salida;
/**
 * @author OLG
 */
public class Muestra {
    public static void muestraEdadyAltura(byte e, float a){
        System.out.println("Tengo " + e + " años");
        System.out.println("Mido " + a);
    }
}
    
```

11

PROGRAMACIÓN

1.- Los métodos estáticos

- Otra opción sería importar solo la clase que voy a utilizar, en este caso la clase Muestra:



```

package edadyaltura2;
import salida.Muestra;

/**
 * @author OLG
 */
public class EdadyAltura2 {

    public static void main(String[] args) {
        byte edad;
        float altura;
        edad = 55;
        altura = 1.75F;
        Muestra.muestraEdadyAltura(edad, altura);
    }
}
    
```

```

package salida;
/**
 * @author OLG
 */
public class Muestra {
    public static void muestraEdadyAltura(byte e, float a){
        System.out.println("Tengo " + e + " años");
        System.out.println("Mido " + a);
    }
}
    
```

12

PROGRAMACIÓN

1.- Los métodos estáticos

- La última opción sería llamar al paquete.clase.método sin utilizar import:



```
package salida;
/**
 * @author OLG
 */
public class Muestra {
    public static void muestraEdadyAltura(byte e, float a){
        System.out.println("Tengo " + e + " años");
        System.out.println("Mido " + a);
    }
}

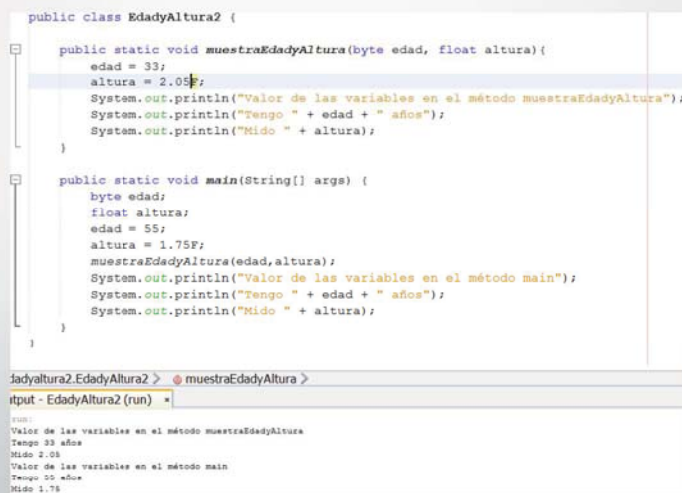
package edadyaltura2;
/**
 * @author OLG
 */
public class EdadyAltura2 {
    public static void main(String[] args) {
        byte edad;
        float altura;
        edad = 55;
        altura = 1.75F;
        salida.Muestra.muestraEdadyAltura(edad, altura);
    }
}
```

13

PROGRAMACIÓN

1.- Los métodos estáticos

- Por último, no olvides lo que vimos sobre el ámbito de las variables, esto es, que aunque cambiemos el valor de una variable dentro de un método, esto no afecta al valor de la variable en otro método distinto:



```
public class EdadyAltura2 {
    public static void muestraEdadyAltura(byte edad, float altura){
        edad = 33;
        altura = 2.05F;
        System.out.println("Valor de las variables en el método muestraEdadyAltura");
        System.out.println("Tengo " + edad + " años");
        System.out.println("Mido " + altura);
    }

    public static void main(String[] args) {
        byte edad;
        float altura;
        edad = 55;
        altura = 1.75F;
        muestraEdadyAltura(edad, altura);
        System.out.println("Valor de las variables en el método main");
        System.out.println("Tengo " + edad + " años");
        System.out.println("Mido " + altura);
    }
}

EdadyAltura2.EdadyAltura2 > muestraEdadyAltura >
Input - EdadyAltura2 (run)
Output:
Valor de las variables en el método muestraEdadyAltura
Tengo 33 años
Mido 2.05
Valor de las variables en el método main
Tengo 55 años
Mido 1.75
```

14

PROGRAMACIÓN