
RELACIÓN DE EJERCICIOS 8 - Métodos

Resolver los siguientes problemas escribiendo el algoritmo con lenguaje Java.

Crear en GitHub un repositorio llamado **UD4-Métodos** para subir cada uno de los ejercicios de la relación.

1. Crea una función con la siguiente cabecera:

```
public static String convierteEnPalotes(int n)
```

Esta función convierte el número *n* al sistema de palotes y lo devuelve en una cadena de caracteres. Por ejemplo, el 470213 en decimal convertido al sistema palotes es:

```
| | | | - | | | | | | - - | | - | - | | |
```

Utiliza esta función en un programa para comprobar que funciona correctamente. Desde la función no se debe mostrar nada por pantalla, solo se debe usar `print` desde el programa principal.

2. Crea una función con la siguiente cabecera:

```
public String convierteEnPalabras(int n)
```

Esta función convierte los dígitos del número *n* en las correspondientes palabras y lo devuelve todo en una cadena de caracteres. Por ejemplo, el 470213 convertido a palabras sería:

cuatro, siete, cero, dos, uno, tres

Utiliza esta función en un programa para comprobar que funciona bien. Desde la función no se debe mostrar nada por pantalla, solo se debe usar `print` desde el programa principal. Fíjate que hay una coma detrás de cada palabra salvo al final.

3. Realiza un programa que pinte un triángulo relleno tal como se muestra en los ejemplos. El usuario debe introducir la altura de la figura. Usar una función para que la implementación sea más sencilla. Por ejemplo, se puede crear una función `linea(char caracter, int repeticiones)` que pinte una línea con el carácter especificado.

Ejemplo 1:

```
Introduzca la altura de la figura: 8
```

```
*****
*****
*****
*****
*****
****
***
**
*

```

RELACIÓN DE EJERCICIOS 8 - Métodos

4. Define la función `convierteArrayEnString` con la siguiente cabecera:

```
public static String convierteArrayEnString(int[] a)
```

Esta función toma como parámetro un array que contiene números y devuelve una cadena de caracteres con esos números. Por ejemplo, si `a = { }`, `convierteArrayEnString(a)` devuelve `""`; si `a = { 8 }`, `convierteArrayEnString(a)` devuelve `"8"`; si `a = { 6, 2, 5, 0, 1 }`, `convierteArrayEnString(a)` devuelve `"62501"`.

5. Define la función `concatena` con la siguiente cabecera:

```
public static int[] concatena(int[] a, int[] b)
```

Esta función toma dos arrays como parámetros y devuelve un array que es el resultado de concatenar ambos. Por ej. si `a = { 8, 9, 0 }` y `b = { 1, 2, 3 }`, `concatena(a, b)` devuelve `{ 8, 9, 0, 1, 2, 3 }`.

6. Crea una biblioteca de funciones para arrays (de una dimensión) de números enteros que contenga las siguientes funciones:

- `generaArrayInt`: Genera un array de tamaño `n` con números aleatorios cuyo intervalo (mínimo y máximo) se indica como parámetro.
- `minimoArrayInt`: Devuelve el mínimo del array que se pasa como parámetro.
- `maximoArrayInt`: Devuelve el máximo del array que se pasa como parámetro.
- `mediaArrayInt`: Devuelve la media del array que se pasa como parámetro.
- `estaEnArrayInt`: Dice si un número está o no dentro de un array.
- `posicionEnArray`: Busca un número en un array y devuelve la posición (el índice) en la que se encuentra.
- `volteaArrayInt`: Le da la vuelta a un array.
- `rotaDerechaArrayInt`: Rota `n` posiciones a la derecha los números de un array.
- `rotaIzquierdaArrayInt`: Rota `n` posiciones a la izquierda los números de un array.

Debes crear un programa que llame a las funciones para probar que funcionan.