

Problema 4: Practicar con arrays y recorridos

Datos personales	
Apellidos:	Cedric
Nombre:	Prieels

1 *Arrays de números reales*

Objetivos

Practicar el recorrido de un array de números.

Descripción

Escribir un método estático Java que reciba como parámetro un array de números reales conteniendo ángulos en grados, y que retorne otro array del mismo tamaño con todos los ángulos del original convertidos a radianes.

Respuesta:

```
public static int[] cambioAGrados(int[] grados) {  
  
    int longitud = grados.length;  
    float radianes[] = new float[longitud];  
  
    for (int i=0; i<longitud; i++) {  
        radianes[i] = Math.toRadians(grados[i]);  
    }  
  
    return radianes[];  
  
}
```

2 *Recorrido parcial de un array*

Objetivos

Practicar el recorrido parcial de un array de números y el paso de pseudocódigo a código.

Descripción

Se supone una clase que tiene como atributo un array x de números reales declarado así:

```
private double[] x;
```

Programación, Curso 2015-2016

Añadir a esta clase un método en Java que codifique el siguiente algoritmo descrito mediante pseudocódigo, que calcula la máxima diferencia entre una casilla del array x y la anterior:

```
método maximaDiferencia() retorna real
  real maxDif=menor número real posible
  para i desde 1 hasta el último índice de x
    dif=valor absoluto(x[i]-x[i-1])
    si dif>maxDif entonces
      maxDif=dif
    fin si
  fin para
  retorna maxDif
fin del método
```

Observar que es un recorrido parcial porque no se empieza en la primera casilla sino en la segunda (que es la de índice 1).

Respuesta:

```
public real maximaDiferencia() {

    double maxDif = -Double.MAX_VALUE;
    double dif = 0;

    for (int i=1; i<x.length; i++) {

        dif = Math.abs(x[i] - x[i-1]);
        if (dif > maxDif) {
            maxDif = dif;
        }

    }

    return maxDif;

}
```

3 *Recorrido de un array de Strings*

Objetivos

Practicar con el uso de un array de Strings y el pseudocódigo

Descripción

Escribir un método al que se le pasa como parámetro un String llamado texto que contiene un texto largo con muchas palabras separadas por espacios en blanco. El método deberá contar y retornar el número de veces que aparece en el texto la palabra indicada en su segundo parámetro, llamado palabra.

Programación, Curso 2015-2016

Para ello el método debe obtener en primer lugar un array de Strings conteniendo en cada casilla una palabra. Se puede usar el método de la clase String llamado `split()`, definido así:

```
String[] split(String separador)
```

Posteriormente, el método deberá hacer un recorrido de todas las casillas del array obtenido y contar aquellas iguales al parámetro `palabra`. Finalmente retornará el valor de esta cuenta.

Nota sobre el método `split()`: Si se usase como separador "-", las palabras deberían ir separadas por caracteres '-'. En cambio, si se usa "\s" esto es un carácter especial que indica cualquier tipo de espacio en blanco (espacios, tabuladores o saltos de línea). Por último, si se usa como separador "\s+", esto indica que las palabras se separan por cualquier secuencia de uno o más espacios en blanco de cualquier tipo, que es lo más apropiado en este caso. Buscar en Internet ejemplos del uso del método `split()`.

Respuesta:

método `contador()` retorna un entero

```
int numero = -1
nuevo array de Strings que contiene las palabras separadas
para i desde 0 hasta el último índice del array
    si la casilla = palabra
        numero aumenta
    fin si
fin para
retorna numero
fin del método
```

```
public int contador(String texto, String palabra) {

    int numero = -1;
    public String[] splitted = texto.split(String "\\s+");

    for (int i = 0; i<splitted.length; i++) {

        if (splitted[i] == palabra) {
            numero ++;
        }

    }

    return numero;

}
```