

# Tema 7 – S6 y S7

## Contenidos:

### 2. Representación y tratamiento de datos mediante un array

- Representación y operaciones básicas... cuando importa la posición (ejemplo del Hospital) y cuando **NO** (ejemplos de la Lista de Enteros y la Agenda de Teléfonos)
- Recorrido de un array: esquemas, ejemplos y ejercicios
- Búsqueda de un dato en un array: esquemas, ejemplos y ejercicios

# Ejercicios ¿De Recorrido o Búsqueda? ¿Combinados?

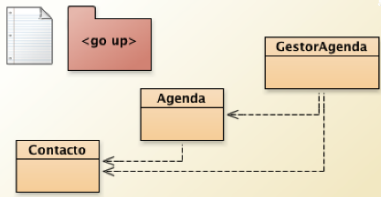


Bluej: ejercicios - Tema 7

## Nº 16 Transparencias:

Con toImpares en mente, **transforma** el método toArray de ListaDeInt en, sucesivamente, los métodos **toArray** y **toLibres** de la clase **Hospital**

**PISTA:** los atributos de Hospital son **Paciente[] elArray**, **C\_P\_D** y **libres**



```
public int[] toArray() {
    int[] res = new int[this.talla];
    for (int i = 0; i < talla; i++) { res[i] = elArray[i]; }
    return res;
}
```

```
public Paciente[] toArray() {
    int ocupadas = C_P_D - this.libres;
    Paciente[] res = new Paciente[ocupadas];
    int j = 0; // índice de res, distinto del índice i de elArray
    for (int i = 1; i < elArray.length && j < ocupadas; i++) {
        if (elArray[i] != null) { res[j++] = elArray[i]; }
    }
    return res;
}
```

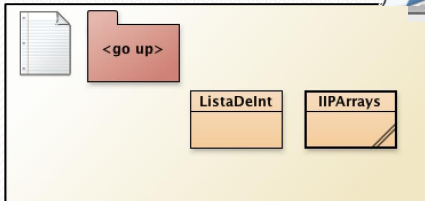
```
public int[] toLibres() {
    int[] res = new int[this.libres];
    for (int i = 1, j = 0; i < elArray.length && j < libres; i++) {
        if (elArray[i] == null) { res[j++] = i; }
    }
    return res;
}
```

# + Ejercicios... ¿De Recorrido o Búsqueda? ¿Combinados?



Bluej: ejercicios - Tema 7

## Ejercicio Nº 12b Transparencias



**Añade** a la clase IIPArrays el método **contarBlancos**, que devuelve el nº de blancos ( ' ' ) que aparecen al final de un array de char v

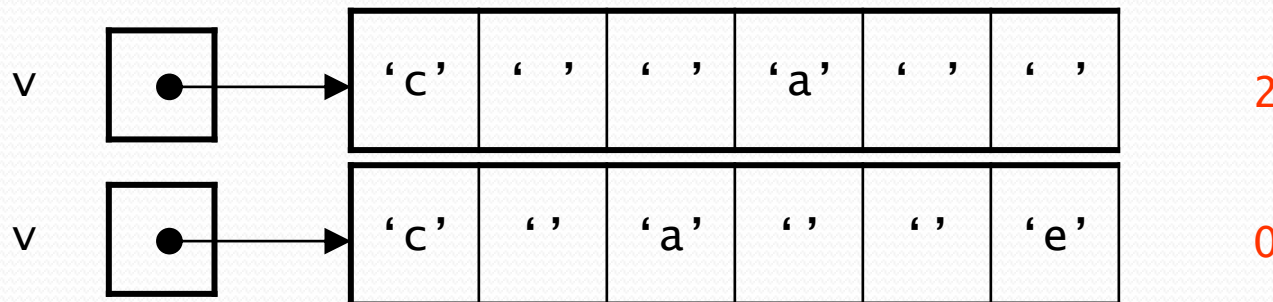
**Preguntas Básicas**, a contestar a partir del ejemplo **Y ANTES** de escribir código:

- 1.- ¿Búsqueda ó Recorrido? **Búsqueda DESCENDENTE**
- 2.- Si es Búsqueda, ¿qué dato se busca?

El primero **Desc.** de v tal que...

$v[i] \neq ' '$

Si está devolver ..., sino **v.length**



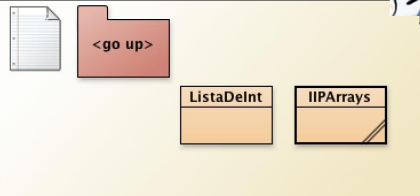
```
public static int contarBlancos(char[] v) {  
    int res = 0;  
    for (int i = v.length - 1; i >= 0; i--) {  
        if (v[i] == ' ') { res++; }  
        else { return res; } // OJO: return SII v[i] != ' '  
    }  
    return v.length;  
}
```

# Ejercicios ¿De Recorrido o Búsqueda? ¿Combinados?



BlueJ: ejercicios - Tema 7

## Nº 18c Transparencias



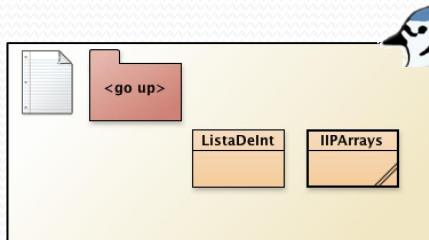
**Añade** a la clase ListaDeInt el método **invertir**, que invierte aquellos elementos de una lista de Enteros que estén situados entre sus posiciones ini y fin,  $0 \leq ini \leq fin < talla()$

### Recorrido “Especlar”

visitando los elementos del array entre ini y fin como en capicua

```
// PRECONDICIÓN:  $0 \leq ini \leq fin < talla()$ 
public void invertir(int ini, int fin) {
    for (int i = ini, j = fin; i < j; i++, j--) {
        int aux = elArray[i];
        elArray[i] = elArray[j];
        elArray[j] = aux;
    }
}
```

# + Ejercicios... ¿De Recorrido o Búsqueda? ¿Combinados?

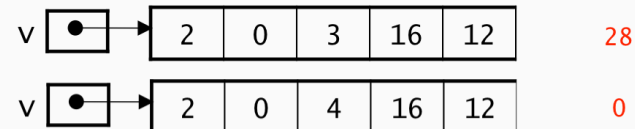


BlueJ: ejercicios - Tema 7

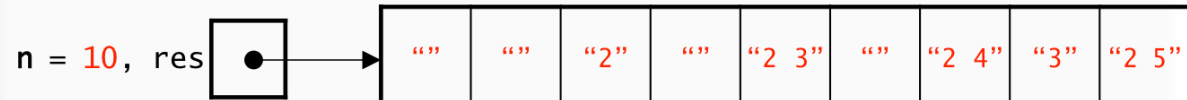
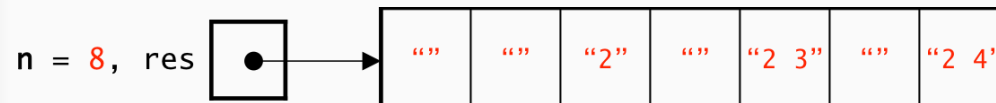
## Nº 19 Transparencias:

**Añade** los siguientes métodos a la clase IIPArrays

- **sumaTras1erImpar**, que devuelve la suma de los elementos que figuran tras el primer impar de un array de Enteros v



- **contarDuplicados**, que devuelve el nº de elementos duplicados (repetidos) en un array de String v
- **divisoresPropiosHasta**, que devuelve un array con las listas (String) de los divisores propios de los números en el intervalo [2, n]



A partir del **19b**,  
COMBINADOS  
**ANIDADOS**

# + Ejercicios... ¿De Recorrido o Búsqueda? ¿Combinados?



BlueJ: ejercicios - Tema 7

Nº 19b Transparencias:



<go up>

ListaDelInt

IIPArrays

**Añade** a la clase IIPArrays el método **contarDuplicados**, que devuelve el número de elementos duplicados (repetidos) en un array de String v

Combinado **ANIDADO**

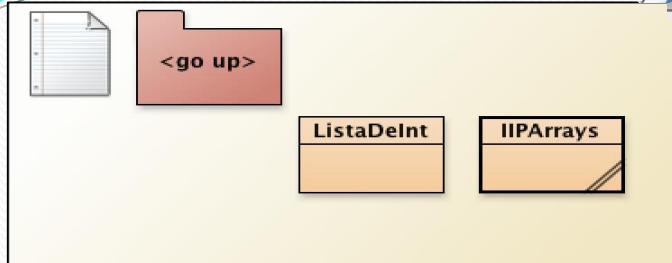
```
public static int contarDuplicados(String[] v)
// Bucle principal: Recorrido Asc. de v (PARA TODO v[i])
int res = 0;
for (int i = 0; i < v.length; i++) {
    // Bucle Anidado: Búsqueda DESDE i+1 del 1er duplicado de v[i]
    // (¿Existe duplicado de v[i], i.e. v[j].equals(v[i])?)
    int j = i + 1;
    while (j < v.length && !v[j].equals(v[i])) { j++; }
    // Resolución de la Búsqueda
    if (j < v.length) { res++; } // hay, al menos, un duplicado
}
return res;
}
```

# + Ejercicios... ¿De Recorrido o Búsqueda? ¿Combinados?



BlueJ: ejercicios - Tema 7

Nº 19c Transparencias:



**Añade** el método **divisoresPropiosHasta** a la clase **IIPArrays**, que devuelve un array con las listas (String) de los Divisores Propios (DP) de los números en el intervalo [2, n]

Combinado **ANIDADO**



**// PRECONDICIÓN:  $n \geq 2$**

```
public static String[] divisoresPropiosHasta(int n)
```

```
    // Declarar e inicializar res:
```

```
    String[] res = new String[n - 2 + 1];
```

```
    // Bucle principal: RECORRIDO del intervalo [2, n] (PARA TODO i)
```

```
    for (int i = 2; i <= n; i++) {
```

```
        // Bucle Anidado: obtener el String res[i], la lista de los DP de i
```

```
        res[i] = "";
```

```
        for (int j = 2; j <= i / 2; j++) {
```

```
            if (i % j == 0) { res[i] += j + " "; }
```

```
        } res[i] = res[i].substring(0, res[i].length() - 1);
```

```
    } return res;
```

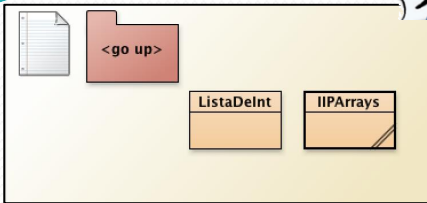
```
}
```

# Ejercicios ¿De Recorrido o Búsqueda? ¿Combinados?



BlueJ: ejercicios - Tema 7

## Nº 13a Transparencias



**Añade** a la clase `ListaDeInt` el método `tresEnOrden`, que devuelve la primera posición de una lista donde se inicia una secuencia ordenada Ascendentemente de 3 Enteros, o -1 si no hay ninguna

**Preguntas Básicas**, a contestar a partir del ejemplo **Y ANTES** de escribir código:

1.- ¿Búsqueda ó Recorrido? Búsqueda **ASC.**

El primero de elArray tal que...

2.- Si es Búsqueda, ¿qué dato se busca?

$\text{elArray}[i] \leq \text{elArray}[i + 1] \leq \text{elArray}[i + 2]$

Si está devolver **i**, sino **-1**

```
public int tresEnOrden() {  
    for (int i = 0; i < talla - 2; i++) {  
        if (elArray[i] <= elArray[i + 1] && elArray[i + 1] <= elArray[i + 2]) {  
            return i;  
        }  
    }  
    return -1;  
}
```



BlueJ: ejercicios - Tema 7

## Nº 17c Transparencias:

**Transforma** el método `tresEnOrden` que se acaba de presentar en el método `tresConsec` de la clase `IIPArrays`, que devuelve la posición donde comienza la 1era subsecuencia de un array de enteros `V` que comprenda, al menos, 3 elementos consecutivos en 3 posiciones consecutivas, o -1 si no existe tal subsecuencia

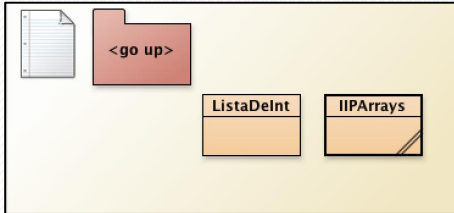


# + Ejercicios... ¿De Recorrido o Búsqueda? ¿Combinados?



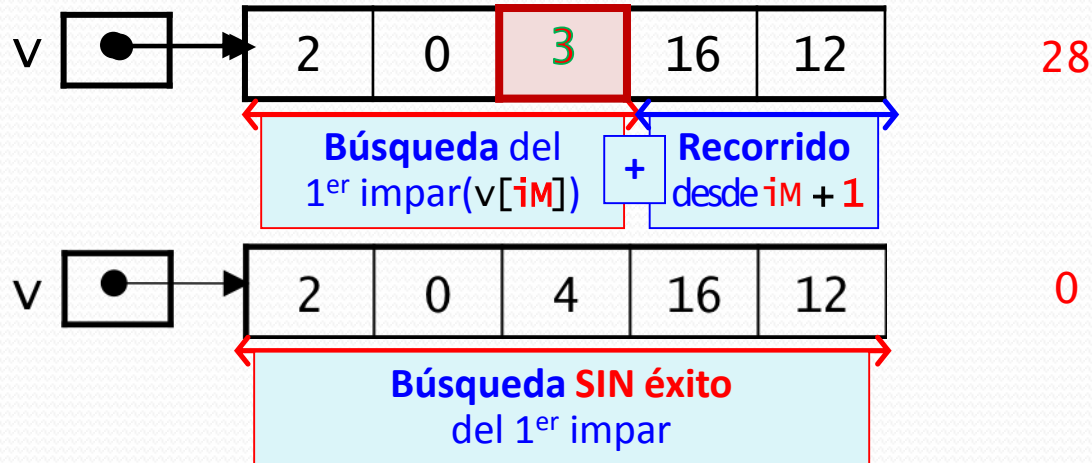
BlueJ: ejercicios - Tema 7

**Nº 19a Transparencias:**



**Añade** a la clase IIPArrays el método **sumaTras1erImpar**, que devuelve la suma de los elementos tras el 1er impar de un array de Enteros v

**COMBINADO** Búsqueda-Recorrido



```
public static int sumarTras1erImpar(int[] v)
// Paso 1: Búsqueda Ascendente del primer impar
int res = 0, i = 0;
while (i < v.length && v[i] % 2 != 1) { i++; }
// Paso 2: Recorrido Asc. DESDE i + 1 hasta v.length - 1
for (int j = i + 1; j < v.length; j++) { res += v[j]; }
return res;
}
```