

# Tema 7 – S5

## Contenidos:

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

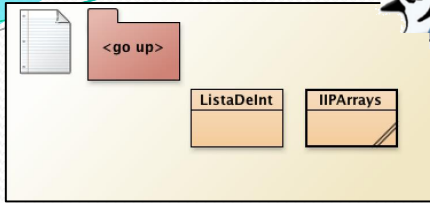
- Representación y operaciones básicas... cuando importa la posición (ejemplo del Hospital) y cuando NO (ejemplos de la Lista de Enteros y la Agenda de Teléfonos)
- Recorrido de un array: esquemas, ejemplos y ejercicios
- Búsqueda de un dato en un array: esquemas, ejemplos y ejercicios
- Métodos para la gestión de un array ORDENADO (**Práctica 7 - Sesión 2**)

# Ejercicios ¿De Recorrido o Búsqueda? ¿Combinados?



BlueJ: ejercicios – Tema 7

## Nº 3 Transparencias:

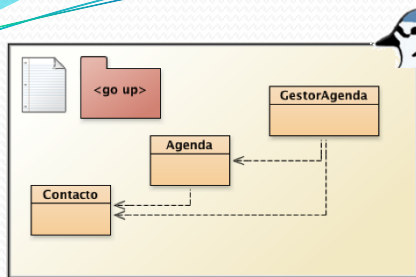


Recorrido **ASC.**

**Implementa** el método **toArray** de la clase **ListaDeInt**, que devuelve un array con los elementos de una Lista de Enteros en el mismo orden en que aparecen en esta (orden de inserción); **nota que**, el array resultado tendrá longitud 0 (array vacío) si la Lista está vacía

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

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



BlueJ: ejercicios - Tema 7

## Ejercicio Nº 10 Transparencias

**Completa** los métodos de Búsqueda de la clase Agenda

**OJO:** en la clase Agenda solo existe un método de búsqueda propiamente dicho: **buscar**; recuerda que insertar, recuperar y eliminar, los otros, solo lo USAN (invocan)

(a) **Estrategia:** si **YA** sabes que es la de Búsqueda, **ANTES** de escribir el código debes responder una pregunta... **¿Qué dato se busca?**

El primer **Contacto** de elArray tal que...

`elArray[i].getNombre().equals(n)`

Si está devolver **i**, sino **-1**

(b) **Traducción a Java**

**PISTA:** “adapta” el código del método buscar de la clase IIPArrays, disponible en la transparencia nº 4 del documento Tema7-S4.pdf. Para ello, recuerda que la clase Agenda tiene 2 atributos de instancia: Contacto[] elArray y talla

```
private int buscar(String n) {
```

```
    for (int i = 0; i < talla; i++) {
        if (elArray[i].getNombre().equals(n)) { return i; }
    }
    return -1;
```

```
}
```

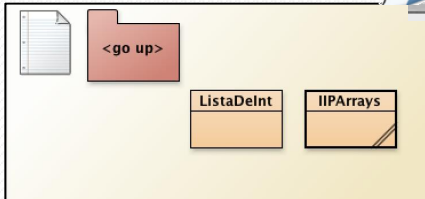
**Revisa** ahora el código de tus métodos insertar, recuperar y eliminar

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



BlueJ: ejercicios – Tema 7

## Ejercicio Nº 12a Transparencias



**Añade** a la clase IIPArrays el método **capicua**, que comprueba si un array de `int` `v` es cap-i-cua (palíndromo)

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

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



```
public static boolean capicua(int[] v) {
```

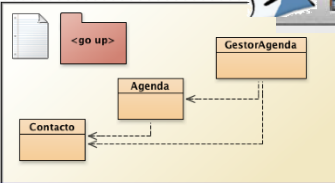
```
}
```

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



BlueJ: ejercicios - Tema 7

## Ejercicio Nº 11 Transparencias



**Añade** a la clase Agenda un método **posMinimo** que devuelva la posición del primer contacto en orden lexicográfico (alfabético) por nombre de una Agenda, o **1er mínimo** de una Agenda según este criterio

Recorrido **ASC.**

**PISTA:** transforma el método posMinimo de la clase IIPArrays

```
public int posMinimo() {  
    int posMin = 0; // Hipótesis inicial: el mínimo  
    String min = elArray[0].getNombre(); // es el 1ero, elArray[0]  
    for (int i = 1; i < talla; i++) {  
        if (elArray[i].getNombre().compareTo(min) < 0) {  
            min = elArray[i].getNombre();  
            posMin = i;  
        }  
    }  
    return posMin;  
}
```



BlueJ: ejercicios - Tema 7

## Ejercicio Nº 14 Transparencias:



**Transforma** el método posMinimo que se acaba de presentar en el **método posMaximo de la clase Hospital**, que devuelve el nº de la última cama ocupada por un paciente sano, o **último máximo** de un Hospital según este criterio

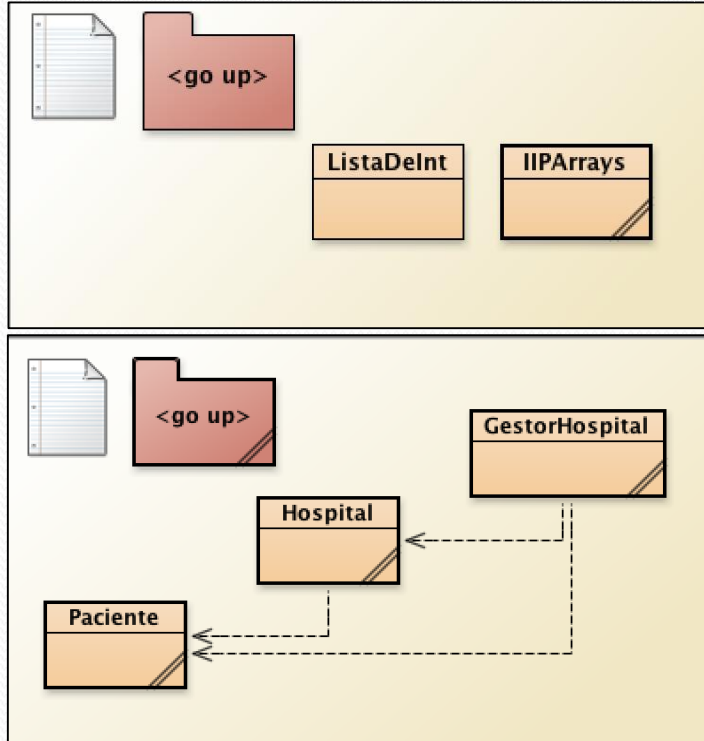
**PISTA:** para mejorar la legibilidad de tu código, añade un método **primeraOcupada()** a la clase Hospital y úsalo en posMaximo

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



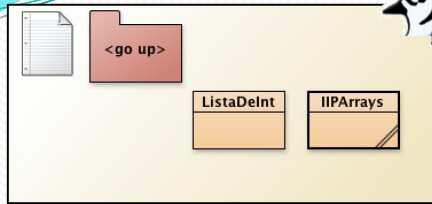
Bluej: ejercicios - Tema 7

**Nº 15 y Nº 16 Transparencias:**



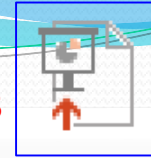
- **Añade** a la clase `ListaDeInt` un método `toImpares`, que devuelve un array con los elementos impares de una lista de enteros en el mismo orden en que aparecen en esta; nota que, el array resultado tendrá longitud 0 (array vacío) si NO hay impares en la Lista
- **Añade** los siguientes métodos a la clase `Hospital`:
  - `toArray`, que devuelve un array con los pacientes de un hospital en el mismo orden en que aparecen en este
  - `toLibres`, que devuelve un array con los números de las camas libres de un hospital en orden ascendente, i.e. en el mismo orden en que aparecen en este; nota que, el array resultado tendrá longitud 0 (array vacío) si NO hay camas libres en el hospital

# Ejercicios ¿De Recorrido o Búsqueda? ¿Combinados?



BlueJ: ejercicios – Tema 7

**Nº 15 Transparencias**

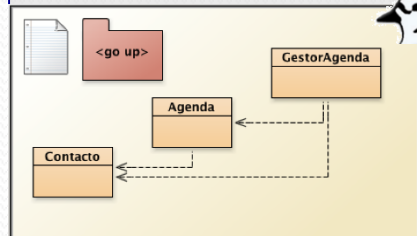


**Añade** a la clase `ListaDeInt` el método **`toImpares`**, que devuelve un array con los elementos impares de una lista de enteros en el mismo orden en que aparecen en esta; nota que, el array resultado tendrá longitud 0 (array vacío) si no hay impares en la lista

**PISTA:** transforma el método `toArray` de la misma clase

```
public int[] toImpares() {
```

```
}
```



BlueJ: ejercicios – Tema 7

**Nº 16 Transparencias:**

**Transforma** el método `toImpares` que se acaba de presentar en, sucesivamente, los **métodos `toArray` y `toLibres`** de la clase `Hospital`

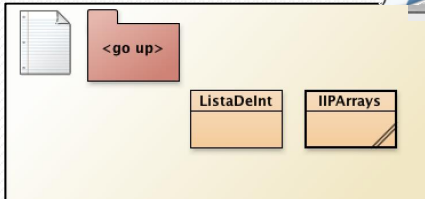
**PISTA:** los atributos de `Hospital` son `Paciente[] elArray`, `C_P_D` y `libres`

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



BlueJ: ejercicios – Tema 7

## Ejercicios 12b y 12c Transparencias



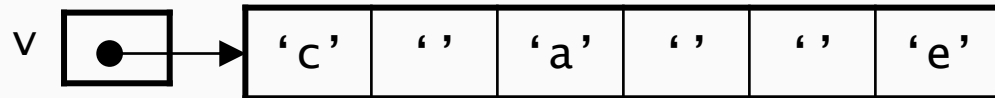
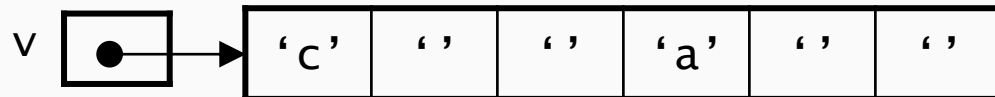
Recorrido AScendente

o

Búsqueda DEScendente

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

- **contarBlancos**, que devuelve el nº de blancos ( ' ' ) que aparecen al final de un array de char v



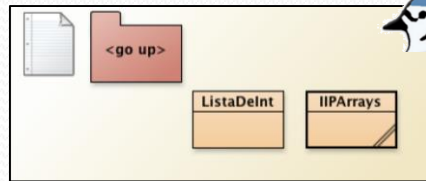
- **ultimoImpar**, que devuelve la posición del último impar de un array de enteros v, o -1 si no hay ninguno



Igual que el 12b



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



BlueJ: ejercicios - Tema 7

**Nº 17 y Nº 18 Transparencias:**

Búsqueda  
DESCendente

Recorrido  
ASCendente

Búsqueda  
ASCendente

Recorrido  
"Especlar"

- **Añade** los siguientes métodos a la clase IIPArrays
  - **ultimaAparicion**, que devuelve la posición de la última aparición de un String s en un array (de String) v, -1 si s no está en v
  - **duplicar**, que duplica el valor de aquellos elementos de un array de enteros v que estén situados entre sus posiciones izq y der,  $0 \leq izq \leq der \leq v.length - 1$
  - **tresConsec**, que devuelve la posición donde comienza la primera subsecuencia de un array de enteros v que comprenda, al menos, tres elementos consecutivos en tres posiciones consecutivas, o -1 si no existe tal subsecuencia
- **Añade** los siguientes métodos a la clase ListaDeInt
  - **desplazarD**, que desplaza una posición hacia la derecha todos los elementos de una lista de Enteros situados entre sus posiciones i y d,  $0 \leq i \leq d < talla() - 1$
  - **desplazarI**, que desplaza una posición hacia la izquierda todos los elementos de una lista de Enteros situados entre sus posiciones i y d,  $0 < i \leq d < talla()$
  - **invertir**, que invierte aquellos elementos de una lista de Enteros que estén situados entre sus posiciones ini y fin,  $0 \leq ini \leq fin < talla()$