

- Actividades propuestas. Realizar las siguientes 5.01, 5.02, 5.03, 5.04, 5.05, 5.06, 5.07 y 5.08.

Actividad propuesta 5.1

Crea tres tablas de cinco elementos: la primera de enteros, la segunda de `double` y la tercera de booleanos. Muestra las referencias en las que se encuentra cada una de las tablas anteriores.

```
1
2
3 package com.mycompany.ap5_1;
4
5 public class Ap5_1 {
6
7     public static void main(String[] args) {
8         int[] a = new int[5];
9         double[] b = new double[5];
10        boolean[] c = new boolean[5];
11        System.out.println(x: a);
12        System.out.println(x: b);
13        System.out.println(x: c);
14    }
15
16 }
17
```

Output - Run (Ap5_1) X

```
skip non existing resourceDirectory C:\Users\abrah\Documents\NetBear
--- compiler:3.11.0:compile (default-compile) @ Ap5_1 ---
Changes detected - recompiling the module! :source
Compiling 1 source file with javac [debug target 21] to target\class
--- exec:3.1.0:exec (default-cli) @ Ap5_1 ---
[I@24d46ca6
[D@4517d9a3
[Z@372f7a8d
-----
BUILD SUCCESS
```

Actividad propuesta 5.2

Construye una tabla de 10 elementos del tipo que desees. Declara diferentes variables de tabla que referenciarán la tabla creada. Comprueba, imprimiendo por pantalla, que todas las variables contienen la misma referencia.

```
1 package com.mycompany.ap5_2;
2
3 public class Ap5_2 {
4
5     public static void main(String[] args) {
6         int[] a = new int[10];
7         int[] b = a;
8         int[] c = a;
9
10        System.out.println(x: a);
11        System.out.println(x: b);
12        System.out.println(x: c);
13    }
14 }
```

Output - Run (Ap5_2) X

Scanning for projects...

-----< com.mycompany:Ap5_2 >-----

Building Ap5_2 1.0-SNAPSHOT
from pom.xml

-----[jar]-----

--- resources:3.3.1:resources (default-resources) @ Ap5_2 ---

--- compiler:3.11.0:compile (default-compile) @ Ap5_2 ---

--- exec:3.1.0:exec (default-cli) @ Ap5_2 ---

[I@24d46ca6
[I@24d46ca6
[I@24d46ca6

BUILD SUCCESS

Actividad propuesta 5.3

Introduce por teclado un número n ; a continuación, solicita al usuario que teclee n números. Realiza la media de los números positivos, la media de los negativos y cuenta el número de ceros introducidos.

```
1 package com.mycompany.ap5_3;
2
3 import java.util.Scanner;
4
5 public class Ap5_3 {
6
7     public static void main(String[] args) {
8         Scanner scanner = new Scanner(System.in);
9         System.out.println(x: "Intoduzca la cantidad de numeros deseada:");
10        int n = scanner.nextInt();
11        int ceros = 0;
12        int pos = 0;
13        int positivos = 0;
14        int neg = 0;
15        int negativos = 0;
16        int []numeros = new int[n];
17        for (int i=0;i<n;i++){
18            System.out.println("Introduzca el " + (i+1) + "º numero");
19            numeros[i] = scanner.nextInt();
20            if(numeros[i] == 0){
21                ceros++;
22            }else if(numeros[i] > 0){
23                pos += numeros[i];
24                positivos++;
25            }else{
26                neg += numeros[i];
27                negativos++;
28            }
29        }
30        if(positivos > 0){
31            System.out.println("Media de los numeros positivos: " + (double)pos/positivos);
32        }else{
33            System.out.println(x: "No hay numeros positivos introducidos para la media");
34        }
35        if(negativos > 0){
36            System.out.println("Media de los numeros negativos: " + neg/negativos);
37        }else{
38            System.out.println(x: "No hay numeros introducidos para la media");
39        }
40        System.out.println("Cantidad de 0 introducidos: " + ceros);
41    }
42 }
43
44 }
```

```
--- exec:3.1.0:exec (default-cli) @ Ap5_3 ---
Intoduzca la cantidad de numeros deseada:
10
Introduzca el 1º numero
6
Introduzca el 2º numero
-9
Introduzca el 3º numero
2
Introduzca el 4º numero
0
Introduzca el 5º numero
0
Introduzca el 6º numero
0
Introduzca el 7º numero
-11
Introduzca el 8º numero
0
Introduzca el 9º numero
10
Introduzca el 10º numero
0
Media de los numeros positivos: 6.0
Media de los numeros negativos: -10.0
Cantidad de 0 introducidos: 5
-----
BUILD SUCCESS
-----
```

Actividad propuesta 5.4

Escribe la función: `int buscar(int t[], int clave)`, que busca de forma secuencial en la tabla `t` el valor `clave`. En caso de encontrarlo, devuelve en qué posición lo encuentra; y en caso contrario, devolverá `-1`.

Clave en el array:

```
1 package com.mycompany.ap5_4;
2
3 public class Ap5_4 {
4
5     public static void main(String[] args) {
6         int pos = buscar(new int[] {10, 25, 9, 4, 547}, clave: 4);
7         if (pos == -1){
8             System.out.println("No se ha encontrado el resultado");
9         }else{
10             System.out.println("El resultado está en la posición: " + pos);
11         }
12     }
13
14     static int buscar(int t[], int clave){
15         int indice = 0;
16         while(indice < t.length && t[indice] != clave){
17             indice++;
18         }
19         if (indice >= t.length){
20             indice = -1;
21         }
22         return indice;
23     }
24 }
```

Output - Run (Ap5_4) X

```
--- resources:3.3.1:resources (default-resources) @ Ap5_4 ---
skip non existing resourceDirectory C:\Users\abrah\Documents\NetBeansProjects\Ap5_4\src\main\resources

--- compiler:3.11.0:compile (default-compile) @ Ap5_4 ---
Changes detected - recompiling the module! :source
Compiling 1 source file with javac [debug target 21] to target\classes

--- exec:3.1.0:exec (default-cli) @ Ap5_4 ---
El resultado está en la posición: 3
BUILD SUCCESS
```

Clave no en el array:

```
1 package com.mycompany.ap5_4;
2
3 public class Ap5_4 {
4
5     public static void main(String[] args) {
6         int pos = buscar(new int[] {10, 25, 9, 4, 547}, clave: 70);
7         if (pos == -1){
8             System.out.println("No se ha encontrado el resultado");
9         }else{
10             System.out.println("El resultado está en la posición: " + pos);
11         }
12     }
13
14     static int buscar(int t[], int clave){
15         int indice = 0;
16         while(indice < t.length && t[indice] != clave){
17             indice++;
18         }
19         if (indice >= t.length){
20             indice = -1;
21         }
22         return indice;
23     }
24 }
```

Output - Run (Ap5_4) X

```
--- compiler:3.11.0:compile (default-compile) @ Ap5_4 ---
Changes detected - recompiling the module! :source
Compiling 1 source file with javac [debug target 21] to target\classes

--- exec:3.1.0:exec (default-cli) @ Ap5_4 ---
No se ha encontrado el resultado
BUILD SUCCESS
```

Actividad propuesta 5.5

Escribe en una función el comportamiento de la inserción ordenada.

```
1 package ap5_5;
2
3 import java.util.Arrays;
4
5
6 public class Ap5_5 {
7
8     public static void main(String[] args) {
9         int tabla[] = insertarOrdenado(new int[]{ 2, 3, 3, 6}, num:4);
10        System.out.println("Insertamos el 4, ahora la tabla es" + Arrays.toString(a: tabla));
11        tabla = insertarOrdenado(tabla, num:1);
12        System.out.println("Insertamos el 1, ahora la tabla es" + Arrays.toString(a: tabla));
13        tabla = insertarOrdenado(tabla, num:4);
14        System.out.println("Insertamos el 4, ahora la tabla es" + Arrays.toString(a: tabla));
15    }
16
17    static int[] insertarOrdenado(int[] tabla, int num){
18        int indice = Arrays.binarySearch(a: tabla, key:num);
19        int[] nTabla = new int[tabla.length + 1];
20        if(indice < 0){
21            indice = -indice - 1;
22        }
23        System.arraycopy(src:tabla, srcPos: 0, dest:nTabla, destPos:0, length: indice);
24        nTabla[indice] = num;
25        System.arraycopy(src:tabla, srcPos: indice, dest:nTabla, indice+1, tabla.length - indice);
26        return nTabla;
27    }
28
29 }
```

ap5_5.Ap5_5 > main >

Output x

Debugger Console x Ap5_5 (run) x

```
run:
Insertamos el 4, ahora la tabla es[2, 3, 3, 4, 6]
Insertamos el 1, ahora la tabla es[1, 2, 3, 3, 4, 6]
Insertamos el 4, ahora la tabla es[1, 2, 3, 3, 4, 4, 6]
BUILD SUCCESSFUL (total time: 0 seconds)
```

Actividad propuesta 5.6

Crea una función que realice el borrado de un elemento de una tabla ordenada.

```
1 package ap5_6;
2
3 import java.util.Arrays;
4
5 public class Ap5_6 {
6
7     public static void main(String[] args) {
8         int tabla[] = new int[]{ 2, 3, 5, 6, 9, 10, 10};
9         System.out.println("Tabla inicial\n" + Arrays.toString(a: tabla));
10        tabla = borrar(tabla, num:5);
11        System.out.println("Tabla tras borrar el 5\n" + Arrays.toString(a: tabla));
12        tabla = borrar(tabla, num:2);
13        System.out.println("Tabla tras borrar el 2\n" + Arrays.toString(a: tabla));
14        tabla = borrar(tabla, num:10);
15        System.out.println("Tabla tras borrar el 10\n" + Arrays.toString(a: tabla));
16    }
17
18    static int[] borrar(int[] tabla, int num){
19        int nueva[] = new int[tabla.length - 1];
20        int indice = Arrays.binarySearch(a: tabla, key:num);
21        if(indice >= 0){
22            System.arraycopy(src:tabla, srcPos: 0, dest:nueva, destPos:0, length: indice);
23            System.arraycopy(src:tabla, indice + 1, dest: nueva, destPos:indice,
24                             tabla.length - indice - 1);
25        }else{
26            nueva = tabla;
27        }
28        return nueva;
29    }
30 }
31
32
```

ap5_6.Ap5_6 > main >

Output x

Debugger Console x

Ap5_6 (run) x

```
run:
Tabla inicial
[2, 3, 5, 6, 9, 10, 10]
Tabla tras borrar el 5
[2, 3, 6, 9, 10, 10]
Tabla tras borrar el 2
[3, 6, 9, 10, 10]
Tabla tras borrar el 10
[3, 6, 9, 10]
BUILD SUCCESSFUL (total time: 0 seconds)
```

Actividad propuesta 5.7

El «número de la suerte» de una persona puede calcularse a partir de sus números favoritos. De entre estos, se seleccionan dos diferentes al azar, que se eliminarán de la lista, pero en su lugar se añade la media aritmética de los dos eliminados a la lista de números favoritos. El proceso se repite hasta que solo quede un número, que resultará el número de la suerte para esa persona. Para calcular bien el número de la suerte es imprescindible que la lista de números se encuentre siempre ordenada.

Escribe una aplicación que solicite al usuario sus números favoritos y calcule su número de la suerte.

```
1 package com.mycompany.ap5_7;
2
3 import java.util.Arrays;
4 import java.util.Scanner;
5
6 public class Ap5_7 {
7
8     public static void main(String[] args) {
9         Scanner scanner = new Scanner(System.in);
10        int num = 0;
11        System.out.println(x: "Introduzca uno a uno sus numeros favoritos ");
12        System.out.println(x: "Pulsar -1 para dejar de introducir");
13        int favoritos[] = new int[]{};
14        while(num != -1){
15            num = scanner.nextInt();
16            favoritos = num != -1 ? insertarOrdenado(tabla: favoritos, num)
17                : favoritos;
18        }
19        System.out.println("Tabla: " + Arrays.toString(a: favoritos));
20        System.out.println("Su numero de la suerte es: " +
21            nSuerte(favoritos));
22    }
23
24    static int nSuerte(int[] favoritos){
25        int media;
26        int [] indices = new int[2];
27
28        while (favoritos.length > 1){
29            if(favoritos.length == 2){
30                favoritos[0] = (favoritos[0] + favoritos[1])/2;
31                favoritos = borrar(tabla: favoritos, favoritos[1]);
32            }else{
33                indices[0] = (int)Math.floor(Math.random()* (favoritos.length-1));
34                indices[1] = (int)Math.floor(Math.random()* (favoritos.length-1));
35                if(indices[0] == indices[1]) continue;
36                Arrays.sort(a: indices);
37                media = (favoritos[indices[0]] + favoritos[indices[1]])/2;
38                favoritos = borrar(tabla: favoritos, favoritos[indices[1]]);
39                favoritos = borrar(tabla: favoritos, favoritos[indices[0]] );
40                favoritos = insertarOrdenado(tabla: favoritos, num: media);
41            }
42        }
43        return favoritos[0];
44    }
45
46    static int[] insertarOrdenado(int[] tabla, int num){
47        int indice = Arrays.binarySearch(a: tabla, key: num);
48        int[] nTabla = new int[tabla.length + 1];
49        if(indice < 0){
50            indice = -indice - 1;
51        }
52        System.arraycopy(src: tabla, srcPos: 0, dest: nTabla, destPos: 0, length: indice);
53        nTabla[indice] = num;
54        System.arraycopy(src: tabla, srcPos: indice, dest: nTabla, indice+1, tabla.length - indice);
55        return nTabla;
56    }
57
58    static int[] borrar(int[] tabla, int num){
59        int nueva[] = new int[tabla.length -1];
60        int indice = Arrays.binarySearch(a: tabla, key: num);
61        if(indice >= 0){
62            System.arraycopy(src: tabla, srcPos: 0, dest: nueva, destPos: 0, length: indice);
63            System.arraycopy(src: tabla, indice +1, dest: nueva, destPos: indice,
64                tabla.length - indice - 1);
65        }else{
66            nueva = tabla;
67        }
68        return nueva;
69    }
70
71 }
72 }
```

```
--- exec:3.1.0:exec (default-cli) @ Ap5_7 ---  
Intoduzca uno a uno sus numeros favoritos  
Pulsar -1 para dejar de introducir  
9  
1  
9  
3  
12  
0  
52  
-1  
Tabla: [0, 1, 3, 9, 9, 12, 52]  
Su numero de la suerte es: 28
```

BUILD SUCCESS

Actividad propuesta 5.8

Comprueba qué produce la comparación con el operador `==` de dos tablas del mismo tipo, la misma longitud y los mismos valores.

```
1 package com.mycompany.ap5_8;
2
3 public class Ap5_8 {
4
5     public static void main(String[] args) {
6         int[] tabla1 = new int[] {10, 25, 9, 4, 547};
7         int[] tabla2 = new int[] {10, 25, 9, 4, 547};
8         System.out.println("Comparamos las dos tablas: " + (tabla1 == tabla2));
9     }
10 }
11
```

Output ×

Debugger Console ×

Run (Ap5_8) ×

```
--- exec:3.1.0:exec (default-cli) @ Ap5_8 ---
Comparamos las dos tablas: false
```

```
-----
BUILD SUCCESS
-----
```

- Actividades de aplicación. Realizar las siguientes 5.11 y 5.17 .

5.11. Realiza la función: `int[] buscarTodos(int t[], int clave)`, que crea y devuelve una tabla con todos los índices de los elementos donde se encuentra la clave de búsqueda. En el caso de que `clave` no se encuentre en la tabla `t`, la función devolverá una tabla vacía.

```

1  package com.mycompany.aa5_11;
2
3  import java.util.Arrays;
4
5  public class Aa5_11 {
6
7      public static void main(String[] args) {
8          int[] t = new int[]{7, 16, 8, 7, 23, 6, 7};
9          int num = 7;
10         System.out.println("En la tabla: " + Arrays.toString(a: t) +
11             " \nel número " + num + " aparece en las posiciones: " +
12             Arrays.toString(a: buscarTodos(t, clave: num)));
13     }
14     static int[] buscarTodos(int[] t, int clave){
15         int[] indices = new int[0];
16         for(int i = 0; i < t.length ; i++){
17             if (t[i] == clave){
18                 indices = insertarOrdenado(tabla: indices, num: i);
19             }
20         }
21
22         return indices;
23     }
24
25
26     static int[] insertarOrdenado(int[] tabla, int num){
27         int indice = Arrays.binarySearch(a: tabla, key: num);
28         int[] nTabla = new int[t.length + 1];
29         if(indice < 0){
30             indice = -indice - 1;
31         }
32         System.arraycopy(src: tabla, srcPos: 0, dest: nTabla, destPos: 0, length: indice);
33         nTabla[indice] = num;
34
35         System.arraycopy(src: tabla, srcPos: indice, dest: nTabla, indice+1, tabla.length - indice);
36         return nTabla;
37     }
38 }

```

```

--- exec:3.1.0:exec (default-cli) @ Aa5_11 ---
En la tabla: [7, 16, 8, 7, 23, 6, 7]
el número 7 aparece en las posiciones: [0, 3, 6]
-----
BUILD SUCCESS
-----

```

5.17. Implementa la función: `int[] suma(int t[], int numElementos)`, que crea y devuelve una tabla con las sumas de los `numElementos` elementos consecutivos de `t`. Veamos un ejemplo, sea `t = [10, 1, 5, 8, 9, 2]`. Si los elementos de `t` se agrupan de 3 en 3, se harán las sumas:

- $10 + 1 + 5$. Igual a 16.
- $1 + 5 + 8$. Igual a 14.
- $5 + 8 + 9$. Igual a 22.
- $8 + 9 + 2$. Igual a 19.

Por lo tanto, la función devolverá una tabla con los resultados: `[16, 14, 22, 19]`.

```
1 package com.mycompany.aa5_17;
2
3 import java.util.Arrays;
4
5 public class Aa5_17 {
6
7     public static void main(String[] args) {
8         int[] t = new int[]{7, 16, 8, 7, 23, 6, 7};
9         int num = 3;
10        System.out.println("En la tabla: " + Arrays.toString(a: t) +
11            " \nel resultado de la suma en grupos de " + num + " es: " +
12            Arrays.toString(a: suma(t, num)));
13    }
14    static int[] suma(int[] t, int num){
15        int[] sumas = new int[0];
16        int suma = 0;
17        for(int i = 0; i + num < t.length + 1; i++){
18            for(int j= i; (j < (i + num)); j++){
19                suma += t[j];
20            }
21            sumas = Arrays.copyOf(original: sumas, sumas.length + 1);
22            sumas[i] = suma;
23            suma = 0;
24        }
25        return sumas;
26    }
27 }
```

Output x

Debugger Console x

Run (Aa5_17) x

```
--- exec:3.1.0:exec (default-cli) @ Aa5_17 ---
En la tabla: [7, 16, 8, 7, 23, 6, 7]
el resultado de la suma en grupos de 3 es: [31, 31, 38, 36, 36]
```

BUILD SUCCESS