

ARRAYS

Ejercicio U4_B11_E1:

```
class Unidad4{
    public static void main(String[] args){
        char[] vocales= new char[5];

        vocales[0] = 'a';
        vocales[1] = 'e';
        vocales[2] = 'i';
        vocales[3] = 'o';
        vocales[4] = 'u';

        for(int i=0;i<5;i++){
            System.out.println("vocales[" + i + "]:" + vocales[i]);
        }
    }
}
```

Ejercicio U4_B11_E2:

```
class Unidad4{
    public static void main(String[] args){
        int[] nums= new int[5];
        int min , max;
        nums[0]=7;
        nums[1]=10;
        nums[2]=3;
        nums[3]=34;
        nums[4]=13;

        min=max=nums[0];
        for(int i=1;i<5;i++){
            if(nums[i]<min)
                min=nums[i];
            if(nums[i]>max)
                max=nums[i];
        }
        System.out.println("El minimo es: "+ min);
        System.out.println("El máximo es: "+ max);

    }
}
```

Ejercicio U4_B11_E3:

```
class Unidad4{
    public static void main(String[] args){
        int[] nums= {7,10,3,34,13};
        int min , max;
        min=max=nums[0];
        for(int i=1;i<5;i++){
            if(nums[i]<min)
                min=nums[i];
            if(nums[i]>max)
                max=nums[i];
        }
        System.out.println("El minimo es: "+ min);
        System.out.println("El máximo es: "+ max);

    }
}
```

Ejercicio U4_B11_E4: imprime la media del siguiente array

```
double[] notas = {8.5,7.0,6.0,9.2};
```

```
class Unidad4{
    public static void main(String[] args){
        double[] notas = {8.5,7.0,6.0,9.2};
        double media=0.0;
        double suma=0.0;
        for(int i=0;i<4;i++)
            suma+=notas[i];
        media=suma/4;
        System.out.println("La media es: "+ media);
    }
}
```

Ejercicio U4_B11_E5:

```
class Unidad4 {
    public static void main(String[] args) {
        boolean[] a={false,false,true,true};
        boolean[] b={false,true,false,true};

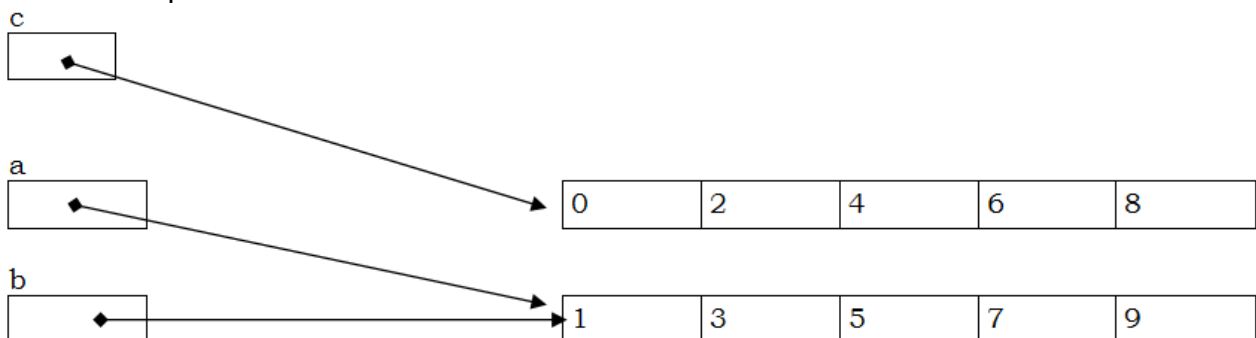
        System.out.println("A\tB\tAorB\tAandB\tAexorB\tAnandB");
        for(int i=0;i<4;i++)
            System.out.println(a[i]+
                                "\t"+b[i]+
                                "\t"+(a[i] || b[i])+
                                "\t"+(a[i] && b[i])+
                                "\t"+(a[i] ^ b[i])+
                                "\t"+!(a[i]&&b[i]));
    }
}
```

Ejercicio U4_B11_E6:

Consigue un aceptado en acepta el reto sea como sea.

Ejercicio U4_B11_E7:

la situación pasa a ser



```
class Unidad4{
    public static void main(String[] args){
        int[] a = {0,2,4,6,8};
        int[] b = {1,3,5,7,9};
        int[] c;
        System.out.println("Array referenciado por a: ");
        for(int i=0;i<5;i++)
            System.out.print(a[i] + " ");
    }
}
```

```

System.out.println("\nArray referenciado por b: ");
for(int i=0;i<5;i++){
    System.out.print(b[i] + " ");
    System.out.println();
    c=a;
    a=b;

    System.out.println("a:" +a);
    System.out.println("b:" +b);
    System.out.println("c:" +c);
    System.out.println("a: " +System.identityHashCode(a));
    System.out.println("b: " +System.identityHashCode(b));
    System.out.println("c: " +System.identityHashCode(c));
}
}

```

Ejercicio U4_B11_E8:

```

class Unidad4{
    public static void main(String[] args){
        int[] a= {1,2,3,4,5};
        int[] b= new int[5]; //o también    int[] b= {0,0,0,0,0};
        for(int i=0;i<5;i++){
            b[i]=a[i];
            System.out.print("a[" + i +"]:" + a[i]);
            System.out.println("    b[" + i +"]:" + b[i]);
        }
    }
}

```

Ejercicio U4_B11_E9

```

class Unidad4{
    public static void main(String[] args){
        int[] a= {1,2,3,4,5};
        int[] b= {0,0,0,0,0};
        //mejor obtener el tamaño de b con a.length
        //int[] b=new int[a.length]
        for(int i=0;i<a.length;i++){
            b[i]=a[i];
            System.out.print("a[" + i +"]:" + a[i]);
            System.out.println(", b[" + i +"]:" + b[i]);
        }
    }
}

```

con método copia con array origen y copia como parámetro

```

class Unidad4{
    static void copia(int[] a,int[] b){
        for(int i=0;i<a.length;i++){
            b[i]=a[i];
        }
    }
    public static void main(String[] args){
        int[] a= {1,2,3,4,5};
        int[] b= {0,0,0,0,0};
        copia(a,b);
        for(int i=0;i<a.length;i++){
            System.out.print("a[" + i +"]:" + a[i]);
            System.out.println(", b[" + i +"]:" + b[i]);
        }
    }
}

```

con método copia con sólo array origen como parámetro y que devuelve(return) un array copiado

```

class Unidad4 {

```

```

static int[] copia(int[] a) {
    int[] copia=new int[a.length];
    for (int i = 0; i < a.length; i++) {
        copia[i] = a[i];
    }
    return copia;
}

public static void main(String[] args) {
    int[] a = {1, 2, 3, 4, 5};
    //no necesario el array lo crea copia()
    //int[] b = {0, 0, 0, 0, 0};
    int[] b;
    b=copia(a);
    for (int i = 0; i < a.length; i++) {
        System.out.print("a[" + i + "]: " + a[i]);
        System.out.println(", b[" + i + "]: " + b[i]);
    }
}
}

```

Ejercicio U4_B11_E10

Con las dos sintaxis anteriores, imprime por pantalla el siguiente array.

```
char[] vocales = {'a', 'e', 'i', 'o', 'u'};
```

```

class Unidad4{
    public static void main(String[] args){
        char[] vocales={'a','e','i','o','u'};
        for(int i=0;i<vocales.length;i++)
            System.out.print(vocales[i]);
        System.out.println();
        for(char v:vocales)
            System.out.print(v);
    }
}

```

Ejercicio U4_B11_E11

Crea un array de 5 amigos de la clase Persona. Pide datos por teclado. Utiliza un bucle para recoger los datos.

```

import java.util.Scanner;
class Persona{
    String nombre;
    int edad;

    public Persona(String nombre, int edad) {
        this.nombre = nombre;
        this.edad = edad;
    }
}

class Unidad4{
    public static void main(String[] args) {
        //un array de amigos
        Scanner teclado = new Scanner(System.in);
        Persona[] amigos = new Persona[5];
        for(int i=0;i<5;i++){
            System.out.print("Nombre: ");
            String nombre=teclado.next();

```

```

        System.out.print("Edad: ");
        int edad=teclado.nextInt();
        amigos[i]=new Persona(nombre,edad);
    }
    System.out.println("Datos introducidos ....");
    for(Persona p:amigos){
        System.out.println(p.nombre + ", "+p.edad);
    }
}
}

```

Ejercicio U4_B11_E12

“cuando parar de leer el array”, simplemente si me encuentro con un elemento persona[i] que valga null es que no tiene persona asociada y por tanto la agenda se acabó. Recuerda que las referencias a objetos se inicializan automáticamente a null por tanto en principio al crear el array persona[0] es null, persona[1] es null etc.

```

import java.util.Scanner;
class Persona{
    String nombre;
    int edad;
    public Persona(String nombre, int edad) {
        this.nombre = nombre;
        this.edad = edad;
    }
}

class Unidad4{
    public static void main(String[] args) {
        //un array de amigos
        Scanner teclado = new Scanner(System.in);
        Persona[] amigos = new Persona[100];

        for(int i=0;i<100;i++){
            String nombre;
            int edad;
            System.out.print("Nombre: ");
            nombre=teclado.next();
            if(nombre.equals("fin")){
                //amigos[i]=new Persona(nombre,0);
                break;
            }
            System.out.print("Edad: ");
            edad=teclado.nextInt();
            amigos[i]=new Persona(nombre,edad);
        }

        System.out.println("Datos introducidos ....");
        for(Persona p:amigos){
            if(p==null)
                break;
            System.out.println(p.nombre + ", "+p.edad);
        }
    }
}

```

Otras formas:

- Otra forma es grabando un falso amigo “fin”. Al recorrer el array si detecto un

- amigo fin es que llegue al final del array
- En lugar de "fin" también podemos usar que para el nombre del último amigo es el String null o el String vacío "".
- Otro enfoque para controlar fin de amigos es utilizar una variable ultimoAmigo que almacena cual es la posición del array que contiene el último amigo.

Ejercicio U4_B11_E13

Cada vez que ingresamos un nuevo amigo, creamos un nuevo array. Esto puede resultar ineficiente o no según el contexto. Para un contexto tipo "agenda de contactos", los elementos del array siempre van a ser pocos y este enfoque no es problemático desde el punto de vista de la eficiencia. Salvo casos concretos no tiene porque ser "ineficiente" crear objetos continuamente, es más, es la solución natural de un programador de POO, digamos que sin un contexto concreto, las dos soluciones anteriores a esta "huelen" a antiguo.

En esta solución el array viejo se copia en el nuevo con arraycopy() que se explica en este boletín. pero si se copia a mano iestá perfecto!.

Como este tipo de problema me resulta novedoso, me puede resultar difícil o desconcertante, entonces, puedo empezar por hacer un pequeño pseudocódigo, como siempre informal y útil para el contexto. Por ejemplo,

```

declaro un array amigos con tamaño 0
repetir hasta que amigo de teclado sea "fin"
    Crear un array amigosNuevo con el tamaño de amigos.length + 1
    copiar viejo en nuevo // y queda al final una posición libre
    meter nuevo amigo al final de amigosNuevo
    amigos=amigosNuevo // ahora la referencia amigos apunta a un nuevo objeto
                                con todos los amigos actualizados

```

Observa como en el seudocódigo mezclé cosas muy concretas como *amigos.length+1* con frases muy genéricas. Esto está bien si te resulta útil, si te aclara la solución. Haz como quieras pero que te valga para algo ya que no tienen sentido los pseudocodigos que te complican la vida.

```

import java.util.Scanner;
class Persona{
    String nombre;
    int edad;

    Persona(String nombre, int edad) {
        this.nombre = nombre;
        this.edad = edad;
    }
    public String toString(){
        return nombre+"", "+edad;
    }
}

class Unidad4{
    public static void main(String[] args){
        Persona[] amigos=new Persona[0];

```

```

Scanner sc= new Scanner(System.in);
System.out.print("Nombre(fin para acabar): ");
String nombre= sc.next();
while(!nombre.equals("fin")){
    System.out.print("edad: ");
    int edad=sc.nextInt();

    Persona[] arrayNuevo=new Persona[amigos.length+1];
    System.arraycopy(amigos,0,arrayNuevo,0,amigos.length);
    arrayNuevo[arrayNuevo.length-1]=new Persona(nombre,edad);
    amigos=arrayNuevo;

    System.out.print("Nombre(fin para acabar): ");
    nombre= sc.next();

}
for(Persona p:amigos){
    System.out.println(p);
}

}

}

```

Ejercicio U4_B3_E14:

```

class Unidad4{
    public static void main(String[] args){
        String[] cadenas = new String[4];
        cadenas[0]="Esta";
        cadenas[1]="es";
        cadenas[2]= "una";
        cadenas[3]="prueba.";
        System.out.println("Array original: ");
        for(int i=0;i<cadenas.length;i++){
            System.out.print(cadenas[i]+ " ");
        }
        cadenas[0]="y esta";
        cadenas[2]="otra";
        System.out.println("\nArray modificado: ");
        for(int i=0;i<cadenas.length;i++){
            System.out.print(cadenas[i] + " ");
        }
    }
}

```

Ejercicio U4_B3_E15:

Con substring fácil

```

class Unidad4{
    public static void main(String[] misArgumentos){
        String palabra=misArgumentos[0];
        System.out.println(palabra);
        for(int i=0;i<palabra.length();i++){
            palabra=palabra.substring(1)+palabra.charAt(0);
            System.out.println(palabra);
        }
    }
}

```