

ARRAYS DE OBJETOS

En los ejercicios y ejemplos que hemos vistos hasta ahora se han creado objetos y hemos trabajado con los objetos creados de forma individual. Esto ha sido posible porque el número de objetos que hemos creado en cada programa ha sido muy pequeño. En aplicaciones reales donde se crean mucha cantidad de objetos lo normal será tenerlos almacenados en un array, en una colección tipo ArrayList o en una base de datos de modo que podamos trabajar con ellos de forma más adecuada.

Vamos a ver con ejemplos como crear un array de objetos y cómo trabajar con ellos.

Ejemplo: Disponemos de la Coche. La clase contiene los atributos matrícula, marca, modelo y precio.

```
public class Coche{
    private String matricula;
    private String marca;
    private String modelo;
    private double precio;

    public String getMatricula() {
        return matricula;
    }

    public void setMatricula(String matricula) {
        this.matricula = matricula;
    }

    public String getMarca() {
        return marca;
    }

    public void setMarca(String marca) {
        this.marca = marca;
    }

    public String getModelo() {
        return modelo;
    }

    public void setModelo(String modelo) {
        this.modelo = modelo;
    }

    public double getPrecio() {
        return precio;
    }

    public void setPrecio(double precio) {
        this.precio = precio;
    }

    @Override
    public String toString() {
        StringBuilder sb = new StringBuilder();
        sb.append("Matrícula: ");
        sb.append(matricula);
        sb.append("\nMarca: ");
        sb.append(marca);
        sb.append("\nModelo: ");
        sb.append(modelo);
        sb.append("\nPrecio: ");
        sb.append(String.format("%.2f", precio));
        return sb.toString();
    }
}
```

Vamos a escribir un programa que cree un array de 5 elementos de tipo Coche.

A continuación se leerán los datos de cada coche por teclado y se introducirán en el array.

Finalmente se mostrará por pantalla cada uno de los coches introducidos en el array.

```
import java.util.Scanner;

public class EjemplolArrayDeObjetos {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        //Creamos un array de 5 elementos de tipo Coche
        Coche[] coches = new Coche[5];

        //Creamos las variables para pedir por teclado los datos de los coches
        String matricula;
        String marca;
        String modelo;
        double precio;

        //Creamos una variable de tipo Coche sin crear el objeto. Solo la variable.
        //Dentro del for será donde vayamos creando los objetos de tipo Coche
        //y guardándolos en el array.

        Coche nuevoCoche;

        //Lectura de los datos de los 10 coches e introducirlos en el array.
        //En cada iteración del bucle for se realiza lo siguiente:
        // - se crea un coche nuevo (un objeto de tipo Coche)
        // - se introducen por teclado los datos del coche y se asignan
        //   al coche creado mediante los métodos set
        // - se almacena el coche en el array

        System.out.println("Introduzca los datos de los coches:");
        for (int i = 0; i < coches.length; i++) {

            //Creamos un coche nuevo y se asigna su dirección
            //de memoria a la variable nuevoCoche
            //en cada iteración del for se creará un objeto de tipo Coche
            nuevoCoche = new Coche();

            //Se introduce por teclado los datos de un coche
            System.out.println("Coche " + (i + 1));
            System.out.print("Matrícula: ");
            matricula = sc.nextLine();
            System.out.print("Marca: ");
            marca = sc.nextLine();
            System.out.print("Modelo: ");
            modelo = sc.nextLine();
            System.out.print("Precio: ");
            precio = sc.nextDouble();
            sc.nextLine(); //limpiar el intro

            //Mediante los métodos set se asignan los valores introducidos
            //por teclado al coche que acabamos de crear
            nuevoCoche.setMatricula(matricula);
            nuevoCoche.setMarca(marca);
            nuevoCoche.setModelo(modelo);
            nuevoCoche.setPrecio(precio);
        }
    }
}
```

```
//Se añade el coche al array
//En la posición i del array guardamos el valor de nuevoCoche
//Realmente en cada posición del array no se guarda el objeto completo.
//En cada posición del array estamos guardando la dirección de memoria
//donde se encuentra el objeto
coches[i] = nuevoCoche;
}

//Mostrar el contenido del array
//Se muestran los datos de cada coche
System.out.println("\nListado de coches");
for (int i = 0; i < coches.length; i++) {
    System.out.println("\nCoche " + (i + 1));
    System.out.println(coches[i]);
}
//Dentro del for, en la instrucción System.out.println(coches[i])
//se invoca al método toString de la clase Coche para cada elemento del array
}
}
```

Ejemplo de ejecución. Lo marcado en gris es lo que se introduce por teclado.

Introduzca los datos de los coches:

Coche 1

Matrícula: 3345KMT

Marca: Renault

Modelo: Clio

Precio: 12500

Coche 2

Matrícula: 1212LJK

Marca: Toyota

Modelo: Yaris

Precio: 20400

Coche 3

Matrícula: 2088LVY

Marca: Nissan

Modelo: X-Trail

Precio: 30150

Coche 4

Matrícula: 0025KBD

Marca: BMW

Modelo: 320D

Precio: 48500

Coche 5

Matrícula: 6002JZH

Marca: Fiat

Modelo: 500

Precio: 15100

Listado de coches

Coche 1

Matrícula: 3345KMT

Marca: Renault

Modelo: Clio

Precio: 12500,00

Coche 2

Matrícula: 1212LJK

Marca: Toyota

Modelo: Yaris

Precio: 20400,00

```
Coche 3
Matrícula: 2088LVY
Marca: Nissan
Modelo: X-Trail
Precio: 30150,00

Coche 4
Matrícula: 0025KBD
Marca: BMW
Modelo: 320D
Precio: 48500,00

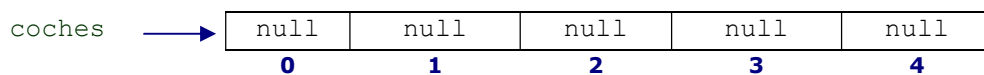
Coche 5
Matrícula: 6002JZH
Marca: Fiat
Modelo: 500
Precio: 15100,00
```

Una explicación de forma gráfica puede ayudarnos a entender mejor cómo funciona el programa.

La instrucción que crea el array es:

```
Coche[] coches = new Coche[5];
```

Se crea un array de 5 elementos. Como se trata de un array para guardar objetos, el valor inicial de cada elemento es null.



A continuación en cada iteración del bucle for se crean los objetos y se añaden al array:

- Primera iteración del for: `i = 0`

La instrucción:

```
nuevoCoche = new Coche();
```

crea un objeto de tipo Coche y asigna su dirección de memoria a la variable nuevoCoche

0x157FC	Coche
	matricula = null
	marca = null
	modelo = null
	precio = 0

```
nuevoCoche = 0x157FC
```

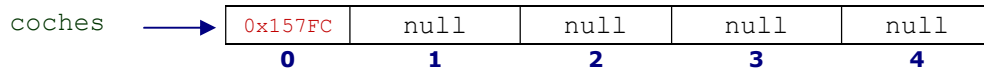
Después se introducen por teclado los datos del coche y mediante los métodos set se asignan a los atributos del coche que se acaba de crear. El contenido del objeto ahora es:

0x157FC	Coche
	matricula = "3345KMT"
	marca = "Renault"
	modelo = "Clio"
	precio = 12500

Por último se guarda el objeto en el array mediante la instrucción:

```
coches[i] = nuevoCoche;
```

Como $i = 0$ en la primera posición del array se guarda el valor de la variable nuevoCoche que es la dirección de memoria donde se encuentra el objeto.

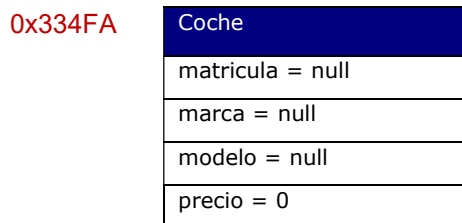


- Segunda iteración del for: $i = 1$

La instrucción:

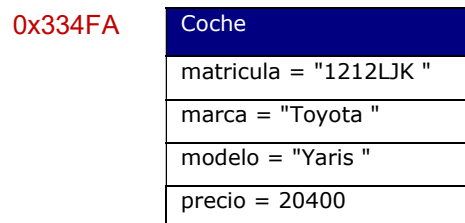
```
nuevoCoche = new Coche();
```

crea un objeto de tipo Coche y asigna su dirección de memoria a la variable nuevoCoche



```
nuevoCoche = 0x334FA
```

Después se introducen por teclado los datos del coche y mediante los métodos set se asignan a los atributos del coche que se acaba de crear. El contenido del objeto ahora es:



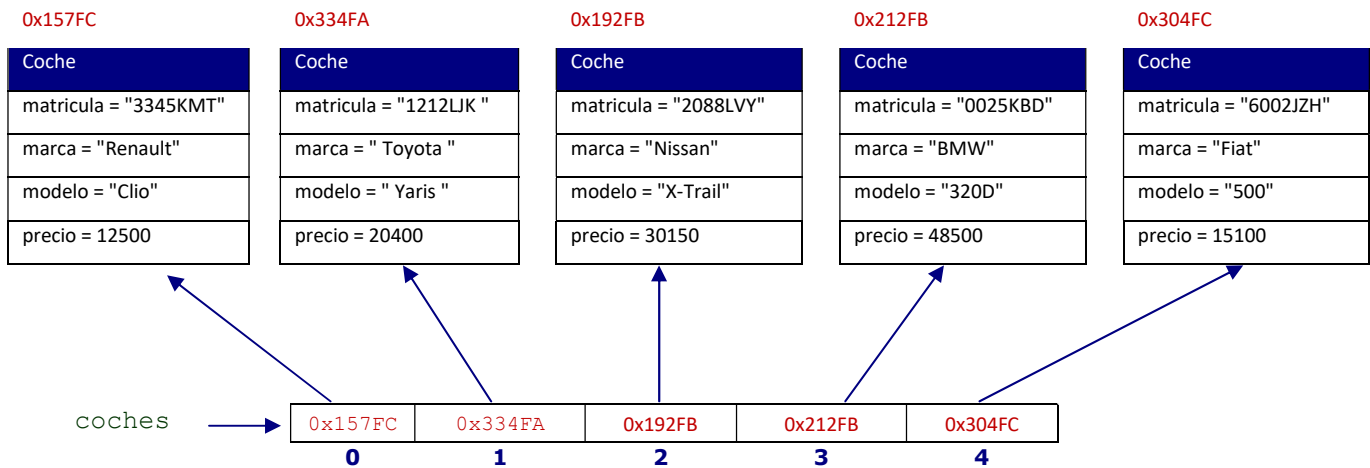
Por último se guarda el objeto en el array mediante la instrucción:

```
coches[i] = nuevoCoche;
```

Como $i = 1$ en la segunda posición del array se guarda el valor de la variable nuevoCoche que es la dirección de memoria donde se encuentra el objeto.



- Para el resto de iteraciones del for el proceso es el mismo. Cuando el for termine tendremos un array de 5 elementos en los que cada elemento tendrá la dirección de memoria de un objeto de tipo Coche.



Cuando después se recorre el array para mostrar los objetos por pantalla, estamos accediendo a cada objeto utilizando la dirección de memoria contenida en cada elemento del array.

En este ejemplo hemos realizado todo el código dentro del método main pero lo habitual será realizar el programa utilizando métodos. En este caso vamos a utilizar dos métodos: un método leerCoches que será el encargado de crear los coches, leer sus datos y añadirlos al ArrayList y otro método mostrarCoches que será el encargado de mostrarlos por pantalla.

El código de la clase principal ahora es este:

```
import java.util.Scanner;

public class Ejemplo2ArraysDeObjetos {
    public static void main(String[] args) {
        //Creamos un array de 5 elementos de tipo Coche
        Coche[] coches = new Coche[5];
        leerCoches(coches); //llamamos al método para leer leer y añadir coches al array
        mostrarCoches(coches); //llamamos al método para mostrar el array
    }

    //Método para leer coches e introducirlos en el array
    public static void leerCoches(Coche[] coches) {
        Scanner sc = new Scanner(System.in);
        //Creamos las variables para pedir por teclado los datos de los coches
        String matricula;
        String marca;
        String modelo;
        double precio;

        //Creamos una variable de tipo Coche sin crear el objeto. Solo la variable.
        //Dentro del for será donde vayamos creando los objetos de tipo Coche
        //y guardándolos en el array.
        Coche nuevoCoche;

        //Lectura de los datos de los 10 coches e introducirlos en el array.
        //En cada iteración del bucle for se realiza lo siguiente:
        // - se crea un coche nuevo (un objeto de tipo Coche)
        // - se introducen por teclado los datos del coche y se asignan
        // al coche creado mediante los métodos set
        // - se almacena el coche en el array
    }
}
```

```
System.out.println("Introduzca los datos de los coches:");
for (int i = 0; i < coches.length; i++) {

    //Creamos un coche nuevo y se asigna su dirección
    //de memoria a la variable nuevoCoche
    //en cada iteración del for se creará un objeto de tipo Coche
    nuevoCoche = new Coche();

    //Se introduce por teclado los datos de un coche
    System.out.println("Coche " + (i + 1));
    System.out.print("Matrícula: ");
    matricula = sc.nextLine();
    System.out.print("Marca: ");
    marca = sc.nextLine();
    System.out.print("Modelo: ");
    modelo = sc.nextLine();
    System.out.print("Precio: ");
    precio = sc.nextDouble();
    sc.nextLine(); //limpiar el intro

    //Mediante los métodos set se asignan los valores introducidos
    //por teclado al coche que acabamos de crear
    nuevoCoche.setMatricula(matricula);
    nuevoCoche.setMarca(marca);
    nuevoCoche.setModelo(modelo);
    nuevoCoche.setPrecio(precio);

    //Se añade el coche al array
    //En la posición i del array guardamos el valor de nuevoCoche
    //Realmente en cada posición del array no se guarda el objeto completo.
    //En cada posición del array estamos guardando la dirección de memoria
    //donde se encuentra el objeto
    coches[i] = nuevoCoche;
}
}

//Método para mostrar todos los coches
public static void mostrarCoches(Coche[] coches) {
    System.out.println("\nListado de coches");
    for (int i = 0; i < coches.length; i++) {
        System.out.println("\nCoche " + (i + 1));
        System.out.println(coches[i]);
    }
}
}
```

También es habitual, sobre todo en aplicaciones grandes, declarar en la clase principal como variables de clase (variables globales) el array y en general todas aquellas variables que se van a utilizar en varios métodos y así nos evitamos tener que estar pasándolas siempre como parámetros a los métodos.

En este ejemplo el array se pasa como parámetro a los métodos leerCoches y mostrarCoches. Podemos declararlo como global y evitar este paso de parámetros. No hay que abusar de esto y a partir de ahora poner todas las variables del programa como globales. Esto solo debemos hacerlo con aquellas variables que se van a utilizar en todos o casi todos los métodos del programa.

El código ahora quedaría así:

```
import java.util.Scanner;

public class Ejemplo3ArraysDeObjetos {
    //Creamos un array de 5 elementos de tipo Coche como variable de clase
    static Coche[] coches = new Coche[5];

    public static void main(String[] args) {
        leerCoches(); //llamamos al método para leer y añadir coches al array
        mostrarCoches(); //llamamos al método para mostrar el array
    }

    //Método para leer coches e introducirlos en el array
    public static void leerCoches() {

        //-----> El código del método leerCoches es el mismo
    }

    //Método para mostrar todos los coches
    public static void mostrarCoches() {
        //-----> El código del método mostrarCoches es el mismo
    }
}
```

Ahora los métodos leerCoches y mostrarCoches no reciben el array como parámetro, al declararlo como variable de clase está disponible en cualquier método de la clase.

Ejemplo: Vamos a modificar la clase Coche y le vamos a añadir un atributo llamado Km de tipo int. Representará el número de kilómetros que tiene el coche.

La clase Coche ahora es esta:

```
public class Coche{

    private String matricula;
    private String marca;
    private String modelo;
    private double precio;
    private int Km;

    public String getMatricula() {
        return matricula;
    }

    public void setMatricula(String matricula) {
        this.matricula = matricula;
    }

    public String getMarca() {
        return marca;
    }

    public void setMarca(String marca) {
        this.marca = marca;
    }

    public String getModelo() {
        return modelo;
    }

    public void setModelo(String modelo) {
        this.modelo = modelo;
    }
}
```



```

    public double getPrecio() {
        return precio;
    }

    public void setPrecio(double precio) {
        this.precio = precio;
    }

    public int getKm() {
        return Km;
    }

    public void setKm(int Km) {
        this.Km = Km;
    }

    @Override
    public String toString() {
        StringBuilder sb = new StringBuilder();
        sb.append("Matrícula: ");
        sb.append(matricula);
        sb.append("\nMarca: ");
        sb.append(marca);
        sb.append("\nModelo: ");
        sb.append(modelo);
        sb.append("\nKm: ");
        sb.append(Km);
        sb.append("\nPrecio: ");
        sb.append(String.format("%.2f", precio));
        return sb.toString();
    }
}

```

Vamos a escribir un programa que crea un array de objetos de tipo Coche.

El tamaño del array se pide por teclado.

A continuación se crearán objetos de tipo Coche y se introducirán en el array.

Después se mostrará:

- Todos los coches introducidos.
- Todos los coches de una marca determinada.
- Todos los coches con menos de un número determinado de Kilómetros.
- El coche con mayor número de Kilómetros.
- Todos los coches ordenados por número de kilómetros de menor a mayor.

```
import java.util.Scanner;
```

```
public class Ejemplo3ArraysDeObjetos {
```

```
    static Coche[] coches; ◀
```

```
    static Scanner sc = new Scanner(System.in);
```

```
    public static void main(String[] args) {
```

```
        leerCoches();
```

```
        System.out.println("\nCoches introducidos:");
```

```
        mostrarCoches();
```

```
        mostrarPorMarca();
```

```
        mostrarPorKm();
```

```
        System.out.println("\nCoche con mayor número de Km: " + mostrarMayorKm());
```

```
        System.out.println("\nCoches ordenados de menor a mayor número de Km");
```

```
        mostrarOrdenadosPorKm();
```

```
    } //fin del método main
```

Se declara un array para guardar objetos de tipo Coche.
El array no se puede crear aún aquí (hacer new ...) porque el tamaño del array se introduce después por teclado.

```
//Método para leer coches e introducirlos en el array
public static void leerCoches() {
    String matricula;
    String marca;
    String modelo;
    int Km;

    int i, N;
    Coche aux;

    //se pide por teclado el número de coches a leer
    do {
        System.out.print("Número de coches? ");
        N = sc.nextInt();
    } while (N < 0);
    sc.nextLine(); //limpiar el intro

    coches = new Coche[N]; //se crea el array coches de tamaño N

    //lectura de N coches
    for (i = 0; i < N; i++) {
        aux = new Coche(); //se crea un coche nuevo

        //introducir por teclado los datos del coche
        System.out.println("Coche " + i);
        System.out.print("Matrícula: ");
        matricula = sc.nextLine();
        System.out.print("Marca: ");
        marca = sc.nextLine();
        System.out.print("Modelo: ");
        modelo = sc.nextLine();
        System.out.print("Número de Kilómetros: ");
        Km = sc.nextInt();
        sc.nextLine(); //limpiar el intro

        //se asignan valores introducidos por teclado a los atributos del objeto
        aux.setMatricula(matricula);
        aux.setMarca(marca);
        aux.setModelo(modelo);
        aux.setKm(Km);

        //se añade el objeto al array
        coches[i] = aux;
    }
}

//Método para mostrar todos los coches
public static void mostrarCoches() {
    for (int i = 0; i < coches.length; i++) {
        System.out.println(coches[i]);
    }
}

//Método para mostrar todos los coches de una marca que se pide por teclado
public static void mostrarPorMarca() {
    String marca;
    System.out.print("Introduce marca: ");
    marca = sc.nextLine();
    System.out.println("Coches de la marca " + marca);
    for (int i = 0; i < coches.length; i++) {
        if (coches[i].getMarca().equalsIgnoreCase(marca)) {
            System.out.println(coches[i]);
        }
    }
}
```

```
//Método para mostrar todos los coches con un número de Km inferior
//al número de Km que se pide por teclado
public static void mostrarPorKm() {
    int Km;
    System.out.print("Introduce número de kilómetros: ");
    Km = sc.nextInt();
    System.out.println("Coches con menos de " + Km + " Km");
    for (int i = 0; i < coches.length; i++) {
        if (coches[i].getKm() < Km) {
            System.out.println(coches[i]);
        }
    }
}

//Método que devuelve el Coche con mayor número de Km
public static Coche mostrarMayorKm() {
    Coche mayor = coches[0]; //tomamos el primer coche como mayor
    for (int i = 0; i < coches.length; i++) {
        if (coches[i].getKm() > mayor.getKm()) {
            mayor = coches[i];
        }
    }
    return mayor;
}

//Método que muestra los coches ordenados por número de Km de menor a mayor
//Para ordenar se utiliza el método de la burbuja
public static void mostrarOrdenadosPorKm() {
    int i, j;
    Coche aux;
    for (i = 0; i < coches.length - 1; i++) {
        for (j = 0; j < coches.length - i - 1; j++) {
            if (coches[j + 1].getKm() < coches[j].getKm()) {
                aux = coches[j + 1];
                coches[j + 1] = coches[j];
                coches[j] = aux;
            }
        }
    }
    mostrarCoches(); //después de ordenar llamamos al método para mostrarlos
}
} //fin de la clase principal
```