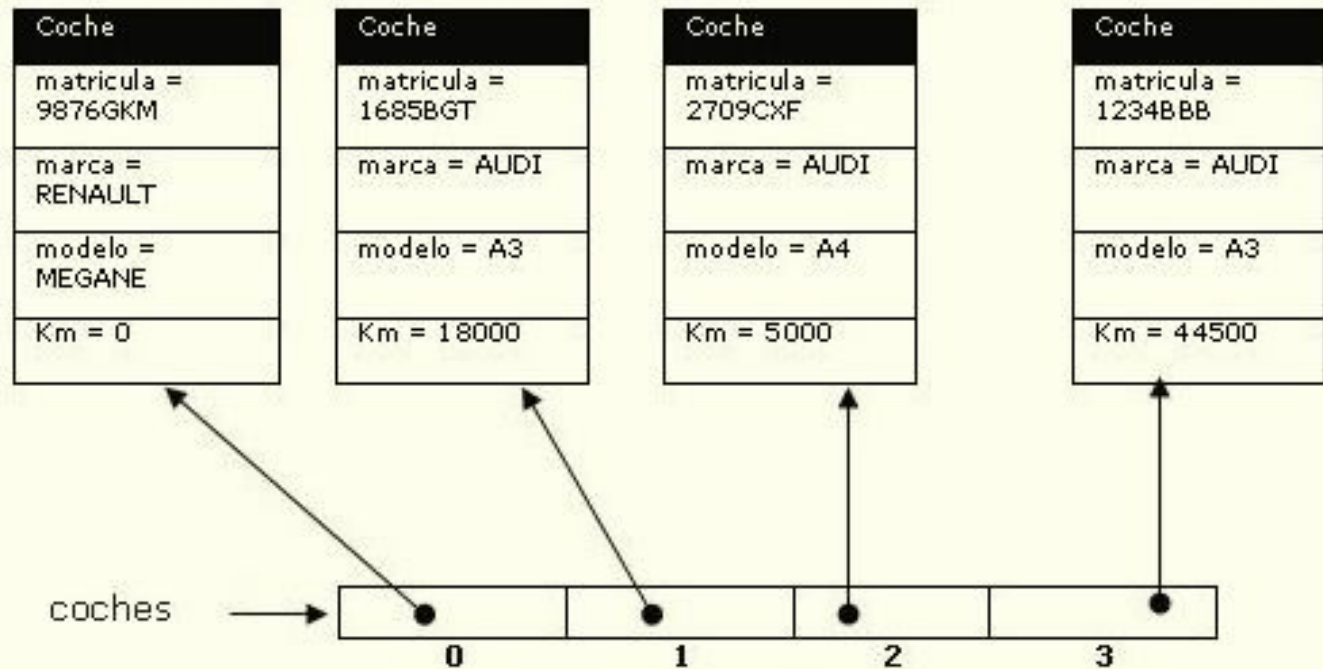


Arreglos de Objetos



Por: Yolanda Martínez Treviño
Ma. Guadalupe Roque Díaz de León

Declaración: arreglo de objetos

Rectangulo **arrRect**[3];

arrRect		
	largo	ancho
0	0	0
1	0	0
2	0	0

Declaración de un
arreglo de objetos

```
// constructor default  
Rectangulo::Rectangulo()  
{  
    largo = 0;  
    ancho = 0;  
}
```

- En cada renglón del arreglo se crea un objeto de la clase Rectangulo.

- Por default cada objeto del arreglo se inicializa con el constructor default.



Métodos **set** para asignar valores a los objetos del arreglo

```
arrRect[indice] ■ setLargo(10);  
arrRect[indice] ■ setAncho(30);
```

Usando los métodos modificadores **set** para asignar los valores

arrRect					
0	<table><tr><th>largo</th><th>ancho</th></tr><tr><td>10</td><td>30</td></tr></table>	largo	ancho	10	30
largo	ancho				
10	30				
1	<table><tr><th>largo</th><th>ancho</th></tr><tr><td>0</td><td>0</td></tr></table>	largo	ancho	0	0
largo	ancho				
0	0				
2	<table><tr><th>largo</th><th>ancho</th></tr><tr><td>0</td><td>0</td></tr></table>	largo	ancho	0	0
largo	ancho				
0	0				

Métodos **get** para acceder los valores del arreglo de objetos

```
cout << "Dimensiones : " <<
```

```
arrRect[indice] ▪ getLargo( ) << "X " <<
```

```
arrRect[indice] ▪ getAncho( );
```

métodos **get** para acceder los valores de los atributos del objeto

arrRect		
	largo	ancho
0	10	30
1	0	0
2	0	0

Pantalla

Dimensiones : 10 X 30

Usando el método **str**

```
cout << "Dimensiones : " << arrRect[indice] • str( ) << endl ;
```

↑
método **str** retorna un string con los atributos del objeto.

arrRect	
0	largo
	ancho
1	largo
	ancho
2	largo
	ancho

Pantalla

Dimensiones : Largo=10
Ancho=30

Usar otros Métodos de la Clase con objetos del arreglo.

```
cout << "Area del rectangulo: " << arrRect[indice] . calculaArea( ) );
```



llamada al método
calculaArea()
de la clase Rectangulo

Pantalla

Area del rectangulo 300

arrRect	
	<div>largo ancho</div>
0	<div>10 30</div>
1	<div>largo ancho</div> <div>0 0</div>
2	<div>largo ancho</div> <div>0 0</div>

Ejercicio:

- Desarrolla las siguientes funciones que reciben como parámetro de entrada el arreglo de objetos de la clase Triangulo e iS indicando la cantidad de elementos a procesar: sintaxis
 - **nombre_funcion(Triangulo arrT[], int iS)**
- **leerArregloTriangulos:** leer los vertices de cada Triangulo
- **desplegarArregloTriangulos:** desplegar el contenido del arreglo de la clase Triangulo
- **calcularPerimetroArea:** calcular y desplegar el perimetro y area de cada uno de los Triangulos del arreglo.
- Menú para que el usuario elija la función que quiere ejecutar en la función main:
 - Dentro de la función **main()**
 - declaración de un arreglo de objetos de la clase Triangulo
 - Haz un ciclo centinela para las opciones de la función menú.

```
int main()
{
    // Declaración de las variables para leer los vertices del triangulo
    double x1, y1, x2, y2, x3, y3;
```

```
// Arreglo de Triangulos
Triangulo arrT[10];

// Calcular area perimetro
calcularAreaPerimetro(arrT, 10);
```

```
// Despliega los 10 elementos del arreglo
muestraArreglo(arrT, 10);

// Despliega los 3 elementos del arreglo
// vemos que el arreglo arrT SE MODIFICO, si CAMBIO,
// esto se debe a que cuando en c++ se envia un arreglo
// el paso de parámetros es por REFERENCIA - 😬
// significa que dentro de la función se esta cambiando
// directamente el argumento correspondiente 😬
```

```
// Leer del teclado 3 triangulos
leerArreglo(arrT, 3);
```

```
muestraArreglo(arrT, 3);
```

```
calcularAreaPerimetro(arrT,10);
```

```
// leer las coordenadas de los 3 puntos
cout << "Ingresa las coordenadas de los 3 vértices x y:";
cin >> x1 >> y1 >> x2 >> y2 >> x3 >> y3;
```

Llamadas a los
funciones
usando el arreglo de
Triangulos

```
// Creacion e inicialización de instancias de la clase Punto:v1,v2,v3
Punto v1{x1, y1}, v2{x2, y2}, v3{x3,y3};
Triangulo t1{v1,v2,v3}, t2{};
```

```
// Desplegar los 3 puntos
cout << "v1 = ";
v1.muestraPunto();
cout << "v2 = ";
v2.muestraPunto();
cout << "v3 = ";
v3.muestraPunto();
```

```
// Desplegar el Triangulo
```

```
// Despliega los 3 elementos del arreglo
// vemos que el arreglo arrT SE MODIFICO, si CAMBIO,
// esto se debe a que cuando en c++ se envia un arreglo
// el paso de parámetros es por REFERENCIA - 😊 ✨ 🙌
// significa que dentro de la función se modificando
// directamente el argumento correspondiente 😊
```

```
// Leer del teclado 3 triangulos
// leerArreglo(arrT, 3);
```

```
// muestraArreglo(arrT, 3);
```

```
// calcularAreaPerimetro(arrT,10);
```

Ejemplo de función que lee un arreglo de Episodios desde el main

```
void leerArregloEpisodio(Episodio arrEpisodio[]){
    string sTitulo;
    int iTemporada;
    double dCalificacion;

    for (int iE = 0; iE < 10; iE++){

        cout << "Ingresa el Titulo:";
        getline(cin, sTitulo);
        arrEpisodio[iE].setTitulo(sTitulo);

        cout << "Ingresa Temporada:";
        cin >> iTemporada;
        arrEpisodio[iE].setTemporada(iTemporada);

        cout << "Ingresa Calificacion(double):";
        cin >> dCalificacion;
        arrEpisodio[iE].setCalificacion(dCalificacion);

        // al leer la calificación no consume el salto de línea
        // el ignore lo elimina.
        cin.ignore();

    }
}
```

Ejemplo de función que despliega un arreglo de Episodios desde el main

```
void desplegarArregloEpisodio(Episodio arrEpisodio[]){  
    for (int iE = 0; iE < 10; iE++){  
        arrEpisodio[iE].imprimir();  
    }  
}
```

Ejercicios:

Dentro del main que usa la clase **Circulo** añade lo siguiente -

- función llamada **muestraArreglo** que recibe un arreglo de objetos de la clase **Circulo** y despliegue el contenido de cada elemento del arreglo de la clase **Circulo** usando el método `str` .
- función llamada **leeArreglo** que recibe un arreglo de objetos de la Clase **Circulo**, la función debe leer los datos de cada elemento del arreglo desde el teclado

Dentro del main añade lo siguiente:

- Declaración de un arreglo de 10 objetos de la clase **Circulo**
- Llamar a la función **leeArreglo** con el arreglo que acabas de declarar
- Llamar a la función **muestraArreglo** con el arreglo que acabas de leer

Ejercicios:

Dentro del main que usa la clase **Rectangulo** añade lo siguiente -

- función llamada **muestraArreglo** que recibe un arreglo de objetos de la clase **Rectangulo**, la función muestra el contenido de cada elemento de un arreglo de la clase **Rectangulo** usando el método imprimir.
- función llamada **leeArreglo** que recibe un arreglo de objetos de la Clase **Rectangulo**, la función debe leer los datos de cada elemento del arreglo desde el teclado

Dentro del main añade lo siguiente:

- Declaración de un arreglo de 10 objetos de la clase **Rectangulo**
- Llamar a la función **leeArreglo** con el arreglo que acabas de declarar
- Llamar a la función **muestraArreglo** con el arreglo que acabas de leer

Ejercicios:

Dentro del main que usa la clase **Evento** añade lo siguiente -

- función llamada **muestraArreglo** que recibe un arreglo de objetos de la clase **Evento**, la función muestra el contenido de cada elemento de un arreglo de la clase **Evento** usando el método imprimir.
- función llamada **leeArreglo** que recibe un arreglo de objetos de la Clase **Evento**, la función debe leer los datos de cada elemento del arreglo desde el teclado

Dentro del main añade lo siguiente:

- Declaración de un arreglo de 10 objetos de la clase **Evento**
- Llamar a la función **leeArreglo** con el arreglo que acabas de declarar
- Llamar a la función **muestraArreglo** con el arreglo que acabas de leer