Programando con B4X

Tema 9 – Clases

Version 1.0, marzo 2021

Autor original: Prokopis Leon

Traducido al español por José Miguel López

Tema 9 – Clases

⏱ **3h**

Lo que los estudiantes aprenderán
• ¿Qué es una clase? • ¿Qué es un objeto? • ¿Qué son los atributos y los métodos? • Crear y usar una clase sencilla en B4J.

A menudo en programación necesitamos referirnos a objetos similares de una única forma. Por ejemplo, los estudiantes en el colegio. Cada estudiante posee un nombre, una dirección, unas clases a las que asistir, títulos, etc. Para gestionar esta información un programador debe organizar estos datos sistemáticamente.

Consejo para el profesor

Las Clases son un tema complejo en programación. Evita entrar en detalles profundos como herencia, encapsulación, etc.

Clases

En el ejemplo de los estudiantes podríamos decir que cada estudiantes posee la siguiente información:

Estudiante	
1	Identificador Escolar
2	Nombre
3	Apellidos
4	Dirección
5	Teléfono
6	Correo electrónico

También necesitamos las siguientes funciones:

- Registrar un nuevo estudiante
- Cambiar la información del estudiante
- Traspasar un estudiante
- Mostrar la información de un estudiante
- Borrar un estudiante

Así pues, para 3 estudiantes, tendríamos los siguientes elementos:

Estudiante 1	
1	125310
2	Augusta
3	Ada Byron
4	Londres
5	37535795
6	ada@lon.uk

Estudiante 2	
1	125311
2	Maria
3	Curie
4	Varsovia
5	678433
6	maria@var.pol

Estudiante 3	
1	125312
2	Muḥammad
3	al-Khwarizmi
4	Jiva
5	646456456
6	algor@jiva.uz

Del ejemplo anterior, podemos concluir que los estudiantes poseen datos y les podemos aplicar funciones similares. El hecho de agrupar los datos de los estudiantes y las funciones que podemos hacer con ellos en un único trozo de código se llama "**clase**". Cada estudiante es un **Objeto** o **Instancia** de la clase. Las variables que caracterizan a un estudiante (nombre, teléfono, dirección, etc.) se llaman **propiedades** y las funciones que podemos aplicarles se llaman **Métodos**.

Recuerda

Llamamos **clase** al agrupamiento de datos y de funciones en un único trozo de código independiente.

Objeto o **Instancia** de la clase son todos los elementos independientes que resultan de usar la clase.

Las variables de un objeto se llaman **propiedades.**

Las funciones que aplicamos a un objeto se llaman **Métodos.**

Las ventajas de usar clases son la flexibilidad en el uso del código, mayor velocidad y facilidad de desarrollo de aplicaciones y la posibilidad de reutilizar el código en otros programas.

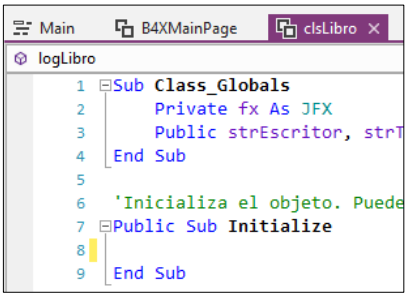
Ejemplo de clase en B4J

Una biblioteca posee un conjunto de libros que presta a los lectores registrados. Cada libro tiene propiedades como el título, el autor, la editorial o el año de publicación. Los libros se puede insertar, mostrar o cambiar.

Crea un aplicación en B4J que implemente la clase Libro con las propiedades y métodos descritos.

Metodología de implementación

1. Crea un aplicación **B4XPages** y ponle de nombre "biblioteca".
2. Elige la opción **Proyecto – Añadir nuevo módulo – Módulo de Clase - Standard Class.**
3. En la ventana, ponle de nombre **clsLibro** (significa: clase Libro)
4. Aparecerá una nueva pestaña llamada **clsLibro**.



5. Dentro de la rutina **Class_Globals**, añade las variables que representarán las propiedades de la clase:
   a. Título del libro
   b. Nombre del autor

c. Editorial
d. Año de publicación

```
Sub Class_Globals
    Private fx As JFX
    Public strEscritor, strTítulo, strAño, strEditorial As String
End Sub
```

6. Finalmente, implementa las subrutinas que ejecutarán los métodos:
    a. Insertar un libro
    b. Mostrar un libro
    c. Cambiar un libro

Consejo para el profesor

Ten cuidado de explicar que todas las variables y métodos que hemos creado son públicos, con lo que los objetos podrán usarlos.

Insertar un libro

Esta subrutina aceptará 4 cadenas de texto como parámetros y las asignará en el orden en que aparecen a las variables **strTítulo, strEscritor, strAño y strEditorial.**

```
14  Public Sub insertarLibro(str1, str2, str3, str4 As String)
15      strTítulo = str1
16      strEscritor = str2
17      strAño = str3
18      strEditorial = str4
19  End Sub
```

Mostrar libro

Esta subrutina mostrará con el comando "Log" las propiedades del libro indicado.

```
21  Public Sub mostrarLibro
22      Log("Título    : " & strTítulo)
23      Log("Escritor  : " & strEscritor)
24      Log("Año       : " & strAño)
25      Log("Editorial : " & strEditorial)
26  End Sub
```

Cambiar libro

Este método aceptará 4 parámetros para cambiar las propiedades del libro en el mismo orden que en el método "insertarLibro":

```
28  Public Sub cambiarLibro(str1, str2, str3, str4 As String)
29      strTítulo = str1
30      strEscritor = str2
31      strAño = str3
32      strEditorial = str4
33  End Sub
```

La rutina "Initialize"

Esta rutina se usa para dar valores iniciales a las variables o bien para realizar alguna otra acción al crear un objeto de la clase:

```
7  Public Sub Initialize
8      strTítulo = ""
9      strEscritor = ""
10     strAño = ""
11     strEditorial = ""
12 End Sub
```

Cómo usar la Clase

Volvemos a la pestaña **B4XMainPage** para usar la clase **clsLibro**.

1. Primero creamos los objetos de la clase clsLibro.

```
7  Sub Class_Globals
8      Private Root As B4XView
9      Private xui As XUI
10     Private libro1 As clsLibro
11     Private libro2 As clsLibro
12 End Sub
```

donde libro1 y libro2 son dos objetos de la clase clsLibro con todas las propiedades y métodos discutidos anteriormente.

Uso de los métodos

```
19 Private Sub B4XPage_Created (Root1 As B4XView)
20     Root = Root1
21     Root.LoadLayout("MainPage")
22
23     libro1.Initialize
24     libro2.Initialize
25
26     libro1.insertarLibro("Neuromante", "William Gibson", "1984", "Ace")
27     libro2.insertarLibro("2001: Una odisea del espacio", "Arthur C. Clarke", "1968", "Ace")
28
29     libro1.mostrarLibro
30     libro2.mostrarLibro
31 End Sub
```

El primer método que se usará en los objetos es el método **Initialize**.

Recuerda

Initialize **no es un método opcional**. Es el primer método que hay que invocar antes de usar un objeto.

El método **insertarLibro** introduce los valores de los dos libros en las propiedades de los objetos.

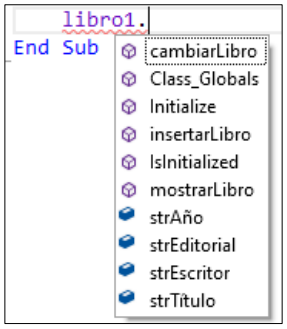
El método **mostrarLibro** muestra las propiedades de cada libro.

```
Título   : Neuromante
Escritor : William Gibson
Año      : 1984
Editorial : Ace
Título   : 2001: Una odisea del espacio
Escritor : Arthur C. Clarke
Año      : 1968
Editorial : Ace
```

Recuerda

Cada propiedad se puede consultar escribiendo el nombre del objeto, seguido de un punto y después el nombre de la propiedad o método. Algunos métodos necesitan parámetros y otros no.

Cada vez que escribas el nombre de un objeto y pulses la tecla con el punto, B4X mostrará una ventana con todas las propiedades y métodos de la clase que hay disponibles. El método **IsInitialized** comprueba si el objeto está inicializado y existe en todas las clase que has creado.



Getters y Setters

En general las propiedades de un objeto deberían ser privadas para evitar que se hagan cambios sin control por parte del objeto. Esto se llama "encapsulación". Para permitir el uso de propiedades privadas, nuestro lenguaje emplea métodos públicos llamados "getters" ("conseguidores") y "setters" ("asinadores") que se construyen añadiendo delante del nombre de la propiedad privada la palabra "get" y "set" respectivamente.

En el ejemplo de los libros, crearemos métodos nuevos y renombraremos el nombre de las variables para evitar "tildes" y "ñ" porque en el nombre de los métodos no se admiten esos caracteres:

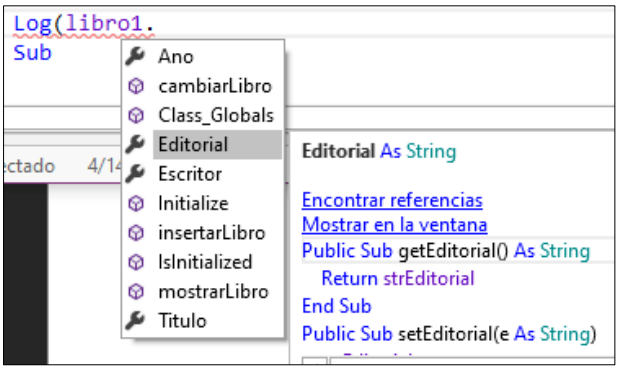
```
Sub Class_Globals
   Private fx As JFX
   Public strEscritor, strTitulo, strAno, strEditorial As String
   Public strEscritor, strTitulo, strAno, strEditorial As String
End Sub

Public Sub getEscritor() As String
   Return strEscritor
End Sub

Public Sub setEscritor(e As String)
   strEscritor = e
End Sub

Public Sub getTitulo() As String
   Return strTitulo
End Sub

Public Sub setTitulo(t As String)
   strTitulo = t
End Sub

Public Sub getAno() As String
   Return strAno
End Sub

Public Sub setAno(y As String)
   strAno = y
End Sub

Public Sub getEditorial() As String
   Return strEditorial
End Sub

Public Sub setEditorial(e As String)
   strEditorial = e
End Sub
```

También sería posible insertar un libro empleando los métodos anteriores:

```
Public Sub insertarLibro(str1, str2, str3, str4 As String)
   setTitulo(str1)
   setEscritor(str2)
   setAno(str3)
   setEditorial(str4)
End Sub
```

Los cambios anteriores al método **insertarLibro** no evitan que se pueda seguir usando como hasta ahora. Por otra parte, al usar los setters se pueden añadir controles adicionales a la hora de cambiar las propiedades.

Finalmente, si quieres usar un setter o un getter tan sólo hay que escribir el nombre de la propiedad:



Por ejemplo, para asignar otro valor al libro1 podemos escribir:

```
   libro1.Escritor = "Wil. Gibson"

   Log(libro1.Escritor)
```

Ejercicios

1. Siguiendo el primer ejemplo del tema, implementa la clase estudiante en B4J con las siguientes propiedades:
   - Identificador escolar
   - Nombre
   - Apellidos
   - Clase
   - Teléfono
   - Correo electrónico

   Y los métodos:
   - Nuevo estudiante
   - Mostrar estudiante
   - Cambiar estudiante
   - Cambiar teléfono

2. Supongamos que un profesor sólo imparte una materia en un colegio. Implementa la clase Curso con las siguientes propiedades:
   a. Tema
   b. Clase
   c. Horas
   d. Profesor

   Y los métodos:
   a. Nuevo Curso
   b. Cambiar Horas
   c. Cambiar Profesor
   d. Mostrar Curso

3. Una tienda vende ordenadores. De cada uno guarda lo siguiente:
   a. El tipo (sobremesa, portátil)
   b. El modelo
   c. El precio
   d. Su CPU (i3, i5, i7, i9)

   Crea una clase que implemente un ordenador con las propiedades anteriores y los métodos nuevoOrdenador, mostrarOrdenador, cambiarCPU y cambiarPrecio. Comprueba que los valores introducidos en CPU y en Tipo son los que aparecen entre paréntesis.