Lesson 9 – Classes

⏱ 3h

What students should know
• What is a Class? • What is an Object? • What are Attributes and Methods? • Create and use simple class with B4J.

Often in programming there is the need to describe similar items in a single way. For example, pupils of a school. Each student is known to have a full name, home address, class to attend, some grades, etc. To monitor and manage this information, a programmer must organize them with systematic way.

Teachers tip

Classes are a tricky subject of programming. Avoid many details and you do not need to go into issues such as inheritance, encapsulation, etc.

Classes

In the previous example with students, we could say that each student has the following information.

Student	
1	Registry Number
2	Name
3	Surname
4	Address
5	Phone
6	Email

At the same time, we need functions such as:

- New Student Registration
- Change Student Information
- Student Transfer
- Show Student Information
- Delete Student

Thus, for three students we could have the following elements:

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

Student 2	
1	125311
2	Maria
3	Curie
4	Warszawa
5	678433
6	maria@wars.pol

Student 3	
1	125312
2	Muḥammad
3	al-Khwarizmi
4	Khwarazm
5	646456456
6	algor@khwa.pe

From the above we conclude that essentially for students we have similar data and similar actions that we can apply to them. Grouping all student data and functions into an independent and single code is called class. Every student Called **Object** or **Instance** of the class. Also, the variables that characterize the student surname, Registry number etc. Called **Properties** and the functions Described **Methods**.

Remember

We call **class** the grouping of data and functions into a single and independent code.

Object or **Instance** of the class are all independent elements that result from the use of the class.

Variables for an object are called **properties.**

The functions applied to an object are called **methods.**

Some of the advantages of using classes are flexibility in the use of code, speed and ease in developing programs and reusing code in other programs.

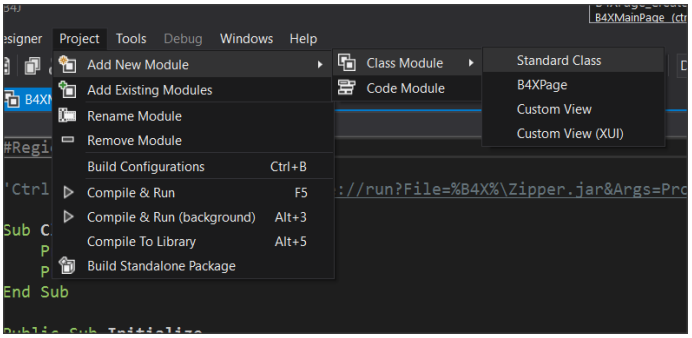
Example of class in B4J

A library has a set of books which it lends to readers who subscribed in the library. Each book has features such as title, author, publisher, year of publication. Also, the Insert Book, Show Book Items, Change Book Items functions apply to each book.

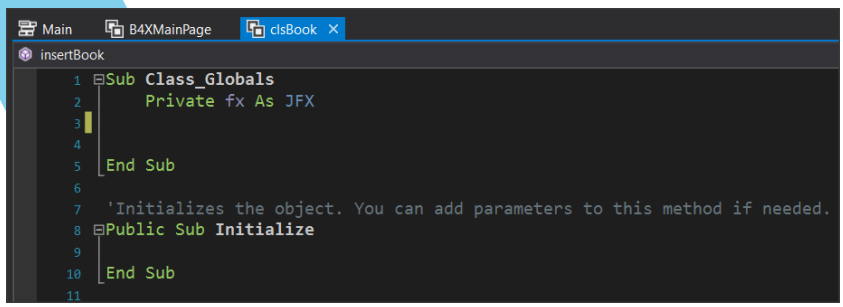
Create an application in B4J that implements the Book class with the properties and methods mentioned above.

Implementation methodology

1. Create a new application B4XPages and give the name "library".
2. From menu Project – Add New Module – Class Module menu select Standard Class.



3. In the dialog box, name clsBook (i.e. class Book)
4. A new code tab named clsBook will be created.



5. Within the Class_Globals routine, add all the variables that will be the properties of the class.
    a. Book Title
    b. Author Name
    c. Publisher
    d. Release Date

```
Sub Class_Globals
    Private fx As JFX
    Public strWriter, strTitle, strYear, strPublisher As String
End Sub
```

6. Finally, you must implement the subprograms that will implement the methods:
    a. Insert a book,
    b. Show Book Items,
    c. Change Book Items

Teachers tip

Be careful to explain that all the variables and methods you create are public so that objects can use them.

Insert a book.

This is a subprogram that accepts 4 string data as parameters and then assigns it in the order that it is inserted into the subprogram into the variables strTitle, strWiter, strYear, strPublisher.

```
Public Sub insertBook(str1, str2, str3, str4 As String)
    strTitle = str1
    strWriter = str2
    strYear = str3
    strPublisher = str4
End Sub
```

Show Book Items.

The subprogram displays with log command the properties of the Books class or in other words the variables that describe the class.

```
Public Sub logBook
    Log("Title : " & strTitle)
    Log("Writer: " & strWriter)
    Log("Year  : " & strYear)
    Log("Publisher: " & strPublisher)
End Sub
```

Change book items.

The method of changing items accepts as parameters new elements for the class and changes the property values of the corresponding properties.

```
Public Sub changeBook(str1, str2, str3, str4 As String)
    strTitle = str1
    strWriter = str2
    strYear = str3
    strPublisher = str4
End Sub
```

The Initialize subprogram.

The Initialize routine is used to give initial values to variables or do any other functions required when creating an object from a class.

```
Public Sub Initialize
    strTitle = ""
    strWriter = ""
    strYear = ""
    strPublisher = ""
End Sub
```

Use Class

Back on tab B4XMainPage it's time to use the clsBook class.

1. First create clsBook objects.

```
Sub Class_Globals
    Private Root As B4XView
    Private xui As XUI
    Private book1 As clsBook
    Private book2 As clsBook
End Sub
```

where book1, book2 are two clsBook objects with all the properties and methods discussed previously.

Use of methods

```
Private Sub B4XPage_Created (Root1 As B4XView)
    Root = Root1
    Root.LoadLayout("MainPage")

    book1.Initialize
    book2.Initialize

    book1.insertBook("Neuromancer", "William Gibson", "1984", "Ace")
    book2.insertBook("2001: A Space Odyssey", "Arthur C. Clarke", "1968", "Ace")

    book1.logBook
    book2.logBook
End Sub
```

The first method to be used in objects is the initialize method.

> **Remember**
>
> Initialize is not optional method. It's the first method you must use before do anything with an object

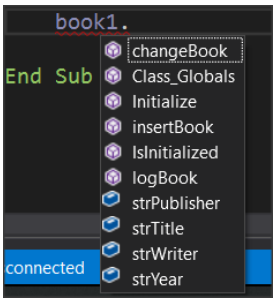
The InsertBook method then enters values for the two books in the object properties.

End the logbook method displays the contents of the properties of each book.

```
Title : Neuromancer
Writer: William Gibson
Year  : 1984
Publisher: Ace
Title : 2001: A Space Odyssey
Writer: Arthur C. Clarke
Year  : 1968
Publisher: Ace
```

> **Remember**
>
> Each property can be called by writing the object name, then a period and the method name. Some methods need parameters to work while others don't.

Each time you write an object name and pressing the period B4X displays a window with all available properties and methods of the class. The IsInitialized method checks whether the object is initialized and exists in all classes that you create.



Getters and Setters

Most of the time the properties of an object are kept private in order to avoid changes beyond what is allowed. This process is called encapsulation. To allow the use of private properties, the language uses public processes called get and set followed by the property name.

In the example of books, additional methods will be created as follows:

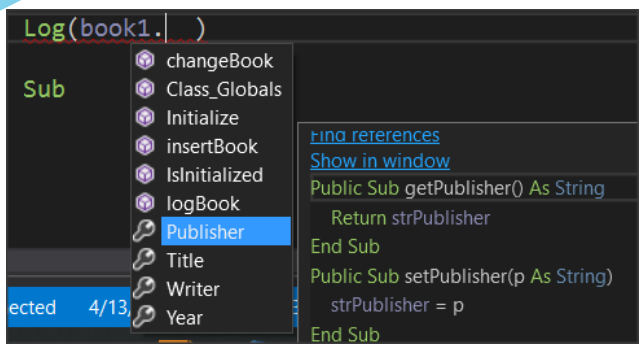
```
Sub Class_Globals
   Private fx As JFX
   Public strWriter, strTitle, strYear, strPublisher As String
   Private strWriter, strTitle, strYear, strPublisher As String
End Sub

Public Sub getWriter() As String
   Return strWriter
End Sub

Public Sub setWriter(w As String)
   strWriter = w
End Sub

Public Sub getTitle() As String
   Return strTitle
End Sub

Public Sub setTitle(t As String)
   strTitle = t
End Sub

Public Sub getYear() As String
   Return strYear
End Sub

Public Sub setYear(y As String)
   strYear = y
End Sub

Public Sub getPublisher() As String
   Return strPublisher
End Sub

Public Sub setPublisher(p As String)
   strPublisher = p
End Sub
```

Also, to insert a book using the insert method can be done:

```
Public Sub insertBook(str1, str2, str3, str4 As String)
   setTitle(str1)
   setWriter(str2)
   setYear(str3)
   setPublisher(str4)
End Sub
```

The previous change to the insertBook method does not exclude its use and by directly assigning the properties of the classes as previously presented. Using set methods, you can integrate additional controls into the data you are about to enter.

Finally, when you want to use the get and set do it by writing the name that follows the get and set.



For example, to set another value to the author of book1 you can simply write:

```
book1.Writer = "Wil. Gibson"

Log(book1.Writer)
```

Exercises

1. In the first example of the course, implement in B4J the Student class with properties:
   - Registry Number
   - Name
   - Surname
   - Class
   - Phone
   - Email

   and methods

   - New Student
   - Show Student
   - Change Class
   - Change Phone

2. If only one teacher teaches a specific course in a school implement the Course class with properties
   a. Lesson
   b. Class
   c. Hours
   d. Professor

   and methods

   a. New Lesson
   b. Change Hours
   c. Change Professor

d. Show Lesson

3. A store has computers for sale. For each computer are recorded:
    a. The type (desktop, laptop),
    b. the model,
    c. its price,
    d. Its cpu (I3, i5, i7, i9)

    Create a class that implements a computer with the above properties and methods that you will create. Be careful to check that the values entered in the type and cpu are those in the parenthesis.