# Lesson 9 – Classes

* What is a Class?
* What is an Object?
* What are Attributes and Methods?
* Create and use simple class with B4J.

What students should know

**3h**

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** | |  | **Student 2** | |  | **Student 3** | |
| 1 | 125310 |  | 1 | 125311 |  | 1 | 125312 |
| 2 | Augusta |  | 2 | Maria |  | 2 | Muḥammad |
| 3 | Ada Byron |  | 3 | Curie |  | 3 | al-Khwarizmi |
| 4 | London |  | 4 | Warszawa |  | 4 | Khwarazm |
| 5 | 37535795 |  | 5 | 678433 |  | 5 | 646456456 |
| 6 | ada@lon.uk |  | 6 | maria@wars.pol |  | 6 | algor@khwa.pe |

From the above we conclude that essentially for students we have identical categories of data and similar actions that we can apply to them. Grouping all student data and functions into an independent and single code creates a Class. Each student record entered becomes an Object or Instance of the Class. Similarly, each Class Object has Properties characterized by variables such as student surname, registry number and so forth. The program instructions that operate on a Class’s Objects are called 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:

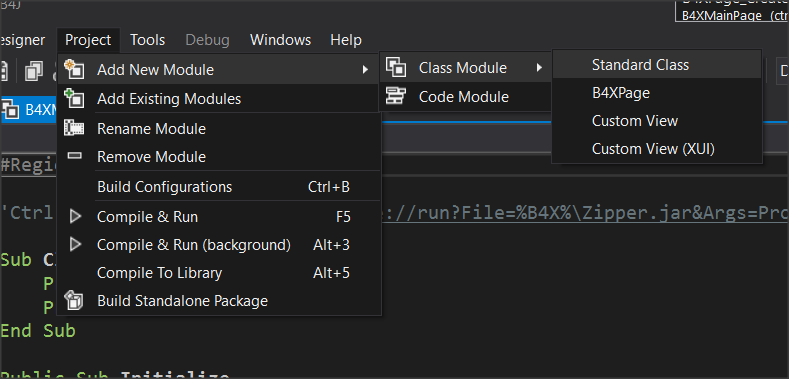
* flexibility in the use of code,
* speed and ease in developing programs and
* reusing code in other programs.

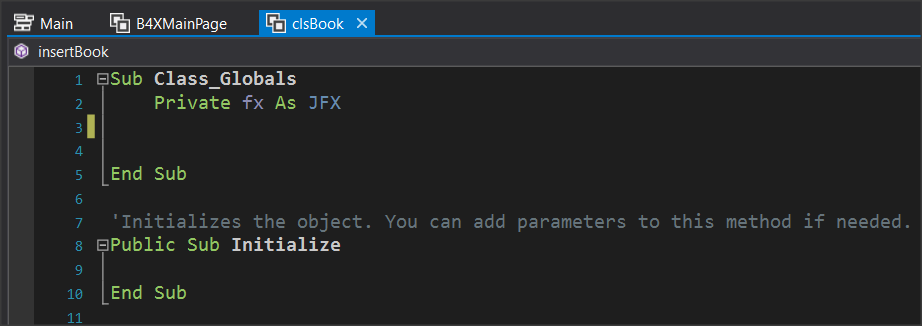
## Example of classes in B4J

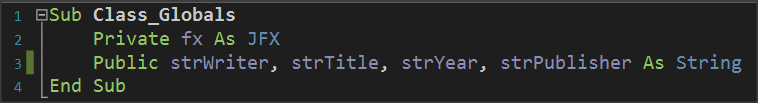
A library has a set of books which it lends to readers who subscribed into the library. Each book has features such as title, author, publisher, year of publication. Also, the functions Insert Book, Show Book Items, Change Book Items apply to each book.

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.



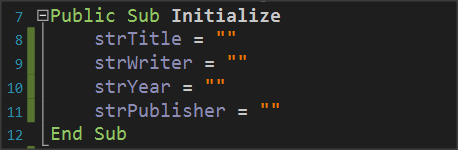
1. Within the Class\_Globals routine, add all the variables that will be the properties of the class.
   1. Book Title
   2. Author Name
   3. Publisher
   4. Year of publication
2. Finally, you must implement the subprograms that will implement the methods:
   1. Insert a book,
   2. Show Book Items,
   3. 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.

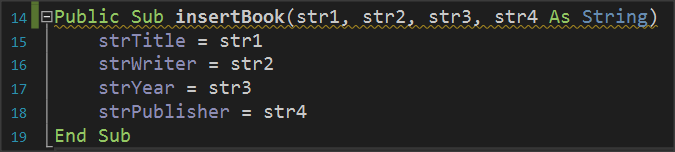


The Initialize subprogram.

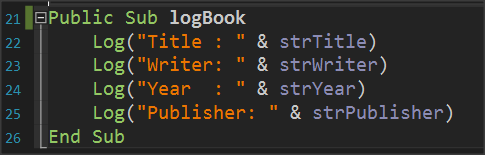
The Initialize routine is used to give initial values to variables or do any other functions required when creating an object from a class.

### Insert a book.

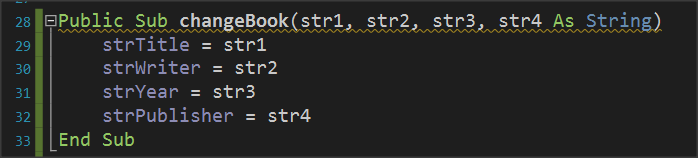
This is a subprogram that accepts 4 string variables as parameters and then assigns them into the variables strTitle, strWriter, strYear, strPublisher.



### 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.

### Change book items.

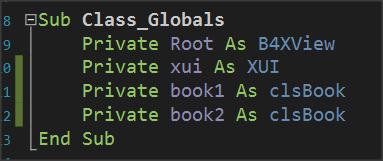
The method of changing items accepts as parameters new elements for the class and changes the property values of the corresponding properties.

### Use Class

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

1. First create clsBook objects.

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



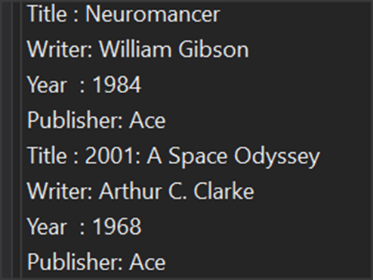
### Use of methods

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 doing anything with an object



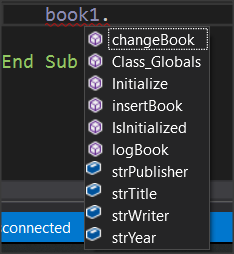
The InsertBook method then enters values for the two books in the object properties.

And, the logBook method displays the contents of the properties of each book.

**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 type an object name and pressing the period B4X displays a popup dialogue 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 called:

**Public Sub** insertBook(str1, str2, str3, str4 **As String**)

**setTitle**(str1)

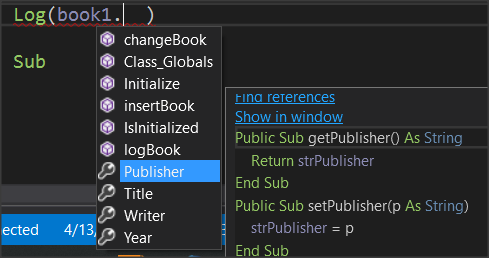
**setWriter**(str2)

**setYear**(str3)

**setPublisher**(str4)

**End Sub**

Using the 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 methods do it by typing 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. Referring to the first example of the lesson, 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

1. If only one teacher teaches a specific course in a school implement the Course class with properties
   1. Lesson
   2. Class
   3. Hours
   4. Professor

and methods

1. New Lesson
2. Change Hours
3. Change Professor
4. Show Lesson
5. A store has an inventory of computers for sale. For each computer, the following are recorded:
   1. The type (desktop, laptop),
   2. the model,
   3. its price,
   4. 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.