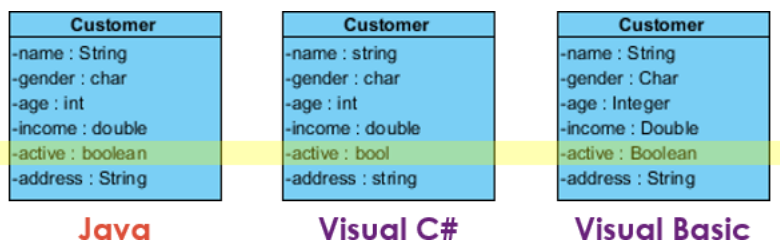




How to Draw ONE Class Diagram for Java, C# and VB?

Written Date : February 16, 2016

Unified Modeling Language (UML) is a generic modeling language that is independent of any particular programming language. Generally speaking, developers should be able to read and understand the diagrams without problem, regardless of the programming language he use. But to make things easier to understand, you can optionally present your [UML class model](#) in specific language. To be specific, data types to use an attributes and operations can be presented in language specific names like 'boolean' in Java and 'bool' in C#.



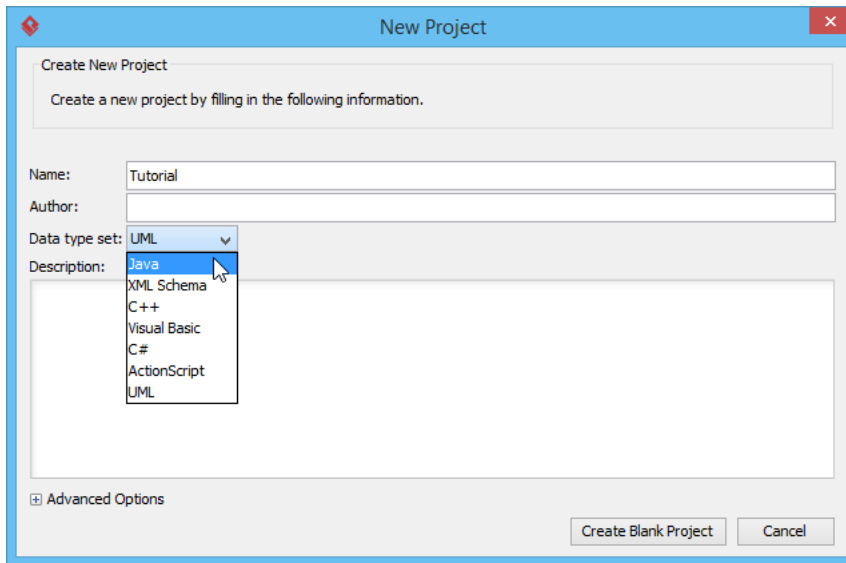
In this tutorial, we will show you how to present a class model in different programming languages.

Creating a project in specific language

In this section we will show you how to create a new project in specific programming language. By doing so you can easily select the language-specific types in constructing a class model. Do not worry if you have not been doing this in your production projects. You can switch between languages any time you want. We will show you how to do this in the next section.

1. Select **Project > New** from the application toolbar.
2. In the **New Project** window, enter *Tutorial* as Name.

- By default, **UML** is selected to be the **Data type set**, meaning that you can use the primitive **UML** data types when constructing your model. Let's say we are going to draw a class diagram for a Java project. Select **Java** to be the **Data type set**.

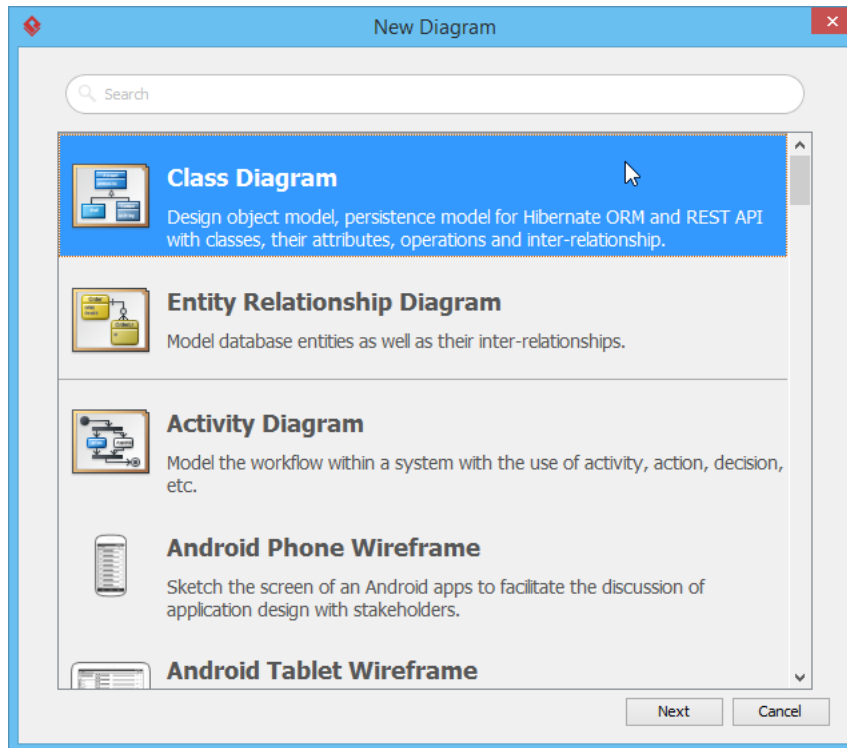


- Click **Create Blank Project**.

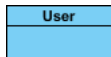
Creating a simple UML class diagram

In this section you will create a class diagram with one class, and several attributes in it. You will be creating the attributes with primitive Java data types.

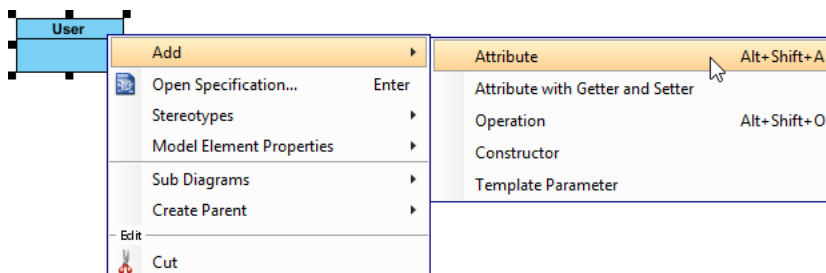
1. Create a UML class diagram first. You can create a class diagram by selecting **Diagram > New** from the application toolbar. Select **Class Diagram** in the **New Diagram** window and then click **Next**. Click **OK** again to create the diagram.



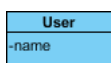
2. Click a class *User*.



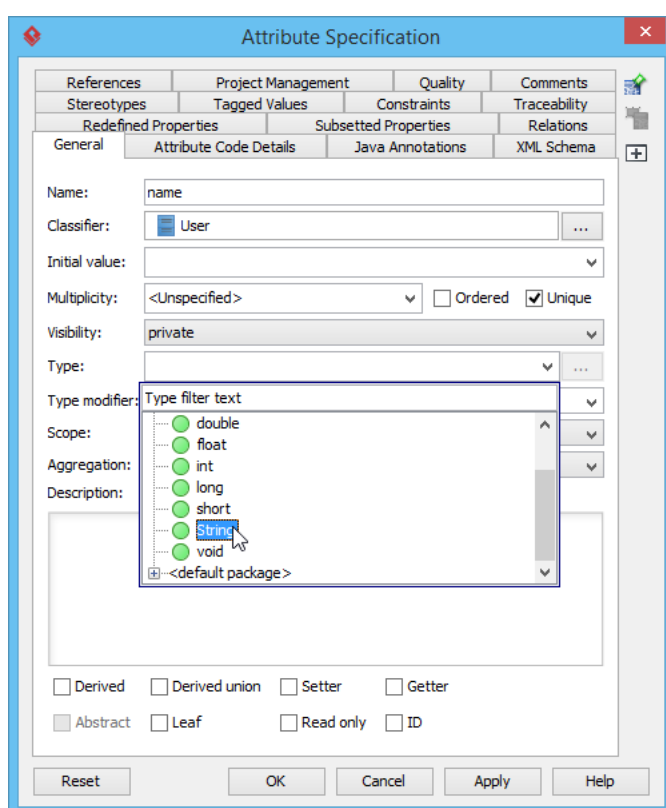
3. Let's add an attribute name into the class. Right click on the class and select **Add > Attribute** from the popup menu.



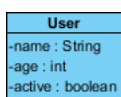
4. *name* is a (Java) String attribute. You can type *name* : *String* to create such an attribute but let's try something different this time. Type *name* and then click on the diagram background to create a typeless attribute.



5. Right click on the attribute and select **Open Specification...** from the popup menu.
6. Click on the drop down menu next to the **Type** field. You can see a list of primitive Java data types available for selection. Now, select **String** and click **OK** to confirm.



7. Now, create two more attributes *age* : *int* and *active* : *boolean*. To save time, you can type the name and data type inline without going through the specification window.

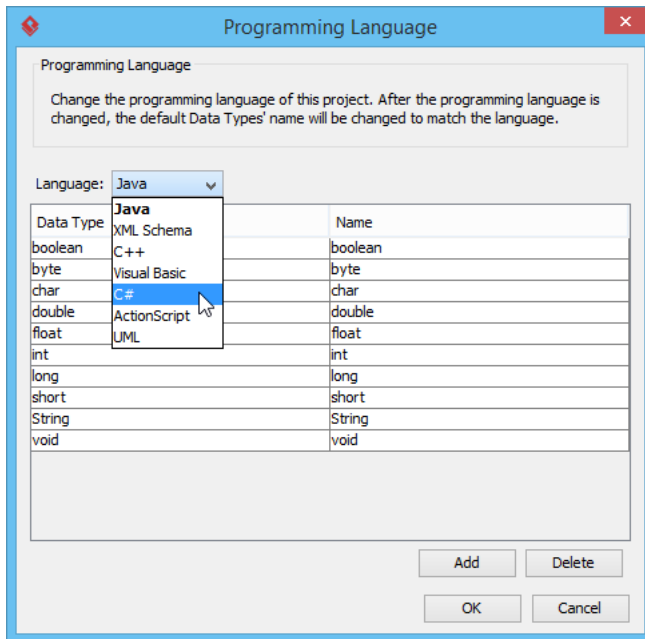


Presenting class model in another programming language

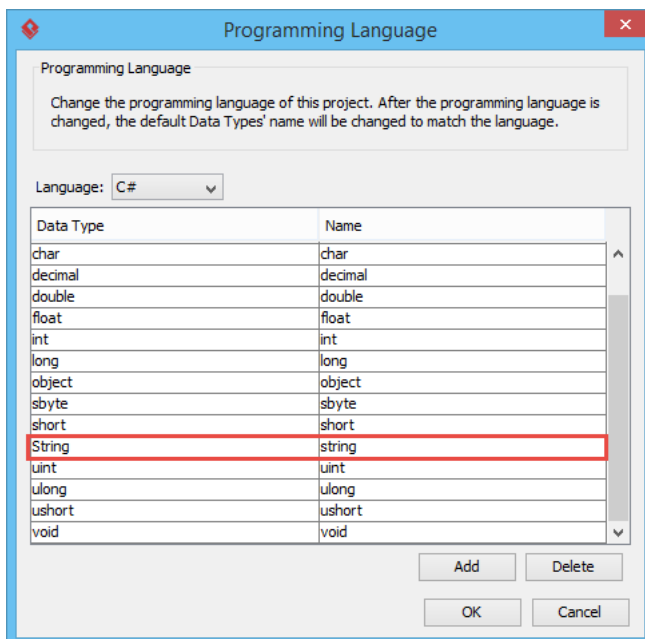
Now you have a class diagram with Java data types used as attribute types. Your Java developers are happy. Let's entertain the C# developers by presenting the data types in C#.

1. Select **Window > Configuration > Configure Programming Language** from the application toolbar.

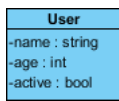
2. The **Programming Language** window shows the currently selected language, its supported data types and their corresponding display names. We will describe in more detail in a minute. Now, change **Language** from **Java** to **C#**.



The list of data types is updated, and is now longer than before. If you scroll you can see some C# types like uint and ulong, which are not available in Java. So how to read the two columns? Let's check the row for String type. The first and second columns are showing String and string respectively. This means that the original String type (available under Java) will be displayed as string by changing the language to C#.



3. Click **OK** to confirm the change of programming language. You can now see the attributes *name* is now showing as a C# string, while *active* is now a C# bool instead of a Java boolean.



Related Links

- [User's Guide - Data type options](#)



[Visual Paradigm home page](https://www.visual-paradigm.com/)
(<https://www.visual-paradigm.com/>)

[Visual Paradigm tutorials](https://www.visual-paradigm.com/tutorials/)
(<https://www.visual-paradigm.com/tutorials/>)