BlueJ IDE

A Quick tour of the BlueJ 3.1.7 IDE

SIUE Computer Science

Last modified: 2016.06.05
IDE key terms

IDE
Integrated Development Environment.

Project
A project in BlueJ is a directory that simply contains all the files that make up the project.

README
The file included with each new project that describes the project.

Toolbox
The panel on the left side of the window. Includes the New Class... and Compile buttons.

Class diagram
The main panel of the IDE that show the class diagrams.

Bench
The area below the main window that holds all the objects.

Code Pad
The area to the right of the bench that is used for entering Java code.

Terminal
The window (console) that displays method calls and/or results of code execution.

Code Editor
The window used to edit and/or add Java code.

Inspector
The inspector dialog that shows the state of an object

Class
A class is represented by an icon in the IDE's class area (the large area above the bench)

Object
An object is represented by an icon that appears in the bench (the area below the class area)
IDE signposts

IDE

Project

README

Toolbox

Class diagram

Bench
signposts continued
Open an existing project

To open an existing project, go to the menu and select **Project | Open Project...** or use the convenient shortcut (For the Mac screen capture below it is ⌘O). Notice that you can also open an existing project.

From the presented **Open Project** dialog choose the project you want to open.
Create a new project

To create a new project, go to the menu and select **Project | New Project...**.

In the presented **New Project** dialog, enter the name for your project in the **File:** text field, and select the directory to house the project. Make note of where you save, since this will make opening the project later a lot easier.
Create a new class

To create a new class, click on the **New Class...** button in the toolbox.

In the presented **Create New Class** dialog, enter the name for the class in the **Class Name:** text field, and select **Class** as the class type. Click **Ok** to create the class.

Stripes indicate the class needs to be compiled.
Add an existing class

To add an existing class, go to the menu and select **Edit | Add Class from File**....

In the presented **Select Java Class to Add** dialog, select the file you would like to add to the project, and click the **Add** button.
Create an object

To create an object, right click on the class an instance of which you want to create, and select one of the available `new <className>()` constructor methods from the pop-up menu.

In the presented **Create Object** dialog, enter the name for the object in the **Name of Instance:** text field, and click the **Ok** button. The newly created instance (object) will now appear in the bench in full red color.
Remove an object

To remove an object from the bench, right click the object you want to remove and select Remove from the pop-up menu.
Inspect an object

To inspect an object, right click the object whose state you want to inspect, and select **Inspect** from the pop-up menu. The inspector dialog will then display the current state of the object.

Alternatively you can double click on the object in the bench.

Further inspection can take place if any of the fields in the object are themselves objects. Notice how the `room` field can be further inspected, since it is a reference to a **String** object.
Call a method from the bench

To call a method on an object that is in the bench, right click the object and select the method you want to execute from the pop-up menu. A Method Result dialog will appear showing the results of the method call.

In the screen capture below the int getId() method was called on the Author instance author1, and the value of the id was returned, currently 0.
Call a method in the Code Pad

To call a method from the code pad - the interactive panel - simply enter the Java code that accomplishes the task.

In the screen capture below, the first statement creates an int variable id, and assigns to it the result of the method call **getId()**, which itself returns the id of the **author1** object. The **author1** object is referred to as the receiving object.
Edit the code

To open up the code editor when you want to edit the class code, right click on the class you want to edit, and select Open Editor from the pop-up menu.

Alternatively you can double click the class icon.

With the editor window displayed you can edit the code by adding new statements, removing statements and even compiling the class itself.
Show results in the Terminal

To show the results of a method call in the Terminal window, select View | Show Terminal or use its shortcut if available.

The terminal window will then be displayed showing the results of the executed code. In the screen capture below, the **author1**'s id is displayed by executing the `System.out.println` method.

```
author1 id: 0
```