

## Chapter 5. Interactions Pane

One of the key distinguishing features of DrJava is its Interactions Pane, which allows you to enter and evaluate Java statements and expressions on the fly. This is remarkably useful for beginning students, who no longer have to learn to write main methods, recompile, and run programs from a command line simply to test how a new class or method behaves. From a teaching standpoint, the Interactions Pane is a very easy way to help students learn to write Java without having to explain the full meaning of syntax like "public static void main", and it also provides an ideal way to perform demonstrations in class. The Interactions Pane can also be used to experiment with your own programs or new libraries, or even to create graphical user interfaces interactively.

### How to Use

The Interactions Pane supports the execution of any valid Java statements as well as the evaluation of Java expressions. Simply define variables and call methods as you would in an ordinary method, or even define new classes and methods and call them interactively. In general, any statement or expression ending without a semicolon will display its result in the Interactions Pane, while those ending with a semicolon will complete without displaying a result. Result objects are displayed using the object's toString() method. Any system output will be displayed in the Interactions Pane in green (as well as in the Console tab), while system errors will be displayed in red by default. Any system input will cause a box to be inserted in the Interactions Pane where you can type what you want System.in to read. This text will be colored purple. These colors can be modified in the "Colors" section in the Preferences window.

Here is a simple interactions session, to demonstrate how the Interactions Pane can be used to experiment with objects or show GUI components.

```
Welcome to DrJava.  
> String s = "Hello World";  
> s  
"Hello World"  
> s.length()  
11  
> import javax.swing.*;  
> JFrame frame = new JFrame("My JFrame");  
> frame.show();  
>
```

### Intelligent Newlines

DrJava parses your input each time Enter is pressed. If it finds that the input is not complete (unmatched braces or a missing semicolon, for example), it will automatically insert a newline, prompting you for more input on the next line. This feature makes declaring loops, methods, and classes very clean.

### Resetting the Interactions Pane

You can reset the Interactions Pane if you wish to start from scratch or if a method call hangs and does not return. Resetting removes any variables from scope and terminates any methods running in the Interactions Pane. To do this, simply choose the "Reset Interactions" command from the Tools menu. This will also reset the Debugger and any JUnit tests that are currently running.

### Running the Main Method

For convenience, DrJava supports calling the main method of a class in the Interactions Pane by simply entering "java" followed by the class name and any arguments.

For example, to run MyClass with args "arg1" and "arg2", type the following into the Interactions Pane:

```
java MyClass arg1 arg2
```

Note that this feature does not support passing special flags or arguments to the JVM itself, as is supported on the command line.

Another shortcut for this feature is the "Run Document's Main Method" command, which can be found in both the Tools menu and the context menu of the document list. This command will simply insert the appropriate "java MyClass" text into the Interactions Pane to run the current class's main method.

### Keyboard Shortcuts

Many actions in the Interactions Pane have keyboard shortcuts for ease of use. Use the Up and Down arrow keys to scroll through a history of the previously entered commands, or Ctrl+B to clear the current command. You can also use Shift+Enter to insert newlines into statements in the Interactions Pane. DrJava also now supports searching backwards through history. To use this feature, type in the first few characters of the command you wish to repeat and hit the Tab key. The last command that matches the characters you typed will appear. Hitting Tab repeated searches farther back, while Shift-Tab will move you forward in the history.

### Setting the Classpath

To interact with any class within the Interactions Pane, it must be included on the Interactions Classpath, which can include more than the user's own classpath. Any class which is opened in the Definitions Pane of DrJava is automatically added to this classpath, but additional classes and directories can be added using the "Extra Classpath" configuration option. (See Configuring DrJava.) The current classpath of the Interactions Pane can be viewed at any time by selecting "View Interactions Classpath" from the context menu.

### Saving the Interactions History

You can save all of your past interactions to a file at any time, using the "Save Interactions History" command in the Tools and popup menus. This command gives you the option to edit any part of the history before saving it, through a separate window that supports editing. By default, up to 500 of the most recent Interaction commands are kept in this history, though this number is configurable. Histories are saved in files with a .hist extension, and they can be later executed in the Interactions Pane with the "Execute Interactions History" command in the Tools menu. Saving and executing histories can be particularly useful for initial setup of an often repeated task, such as importing several packages and initializing frequently used variables. To help manage the history, a "Clear Interactions History" command is also provided in the Tools menu.

### Loading a History File as a Script

You can load a history file as a script that can be executed one line at a time, using the "Load Interactions History as Script" command in the Tools and popup menus. A panel will appear on the right side of the Interactions Pane with buttons that allow you to display the previous interaction, execute the current interaction, display the next interaction, and close the panel. This feature is useful during presentations because you can step through a series of interactions that has been prepared in advance, allowing the audience to see the result of each interaction.

### Lifting Interactions to Definitions

One common use of the Interactions Pane is to test a line of code intended for a program, even before it is written in the program itself. For example, this can be the case

when experimenting with method calls to determine their results. In this situation, it is convenient to copy a working line from the Interactions Pane into the Definitions Pane. This can be done quickly with the "Lift Current Interaction to Definitions" command in the Tools menu, which simply copies the text at the current prompt and pastes it at the cursor position in the Definitions Pane.

### **Context Menu**

The Interactions Pane has a context menu, which can be used by right-clicking in the pane. (Mac users should use Ctrl+Click or Option+Click.) This menu provides shortcuts to useful commands for the Interactions Pane, including cut, copy, and paste, as well as resetting the Interactions Pane, executing, loading, and saving history files, viewing the current classpath, and copying the current interaction to the Definitions Pane.

### **Tiger Features**

DrJava provides support for Tiger (JDK 1.5) features in the interactions pane. These include enum types, static import, for each statements, methods with variable arguments, generics, and autoboxing. Note that you must be running Java 2, version 1.5.0 or later to use 1.5 features in the Interactions Pane. Also, DrJava does not currently type check generics, so while they may be used in the interactions pane without a syntax error, we do not yet provide full support for type checking of generics.