Java SE 9

The Java Shell

LUXOFT TRAINING

LXFT LISTED NYSE

Summary

Provide an **interactive tool** to evaluate declarations, statements, and expressions in Java.

Motivation

Immediate feedback is important when learning a programming language.

The number one reason schools cite for moving away from Java as a teaching language is that other languages have a **"REPL"** and have far lower bars to an initial "Hello, world!" program.

Motivation

A **Read-Eval-Print Loop** is an interactive programming tool which loops, continually reading user input, evaluating the input, and printing the value of the input or a description of the state change the input caused.

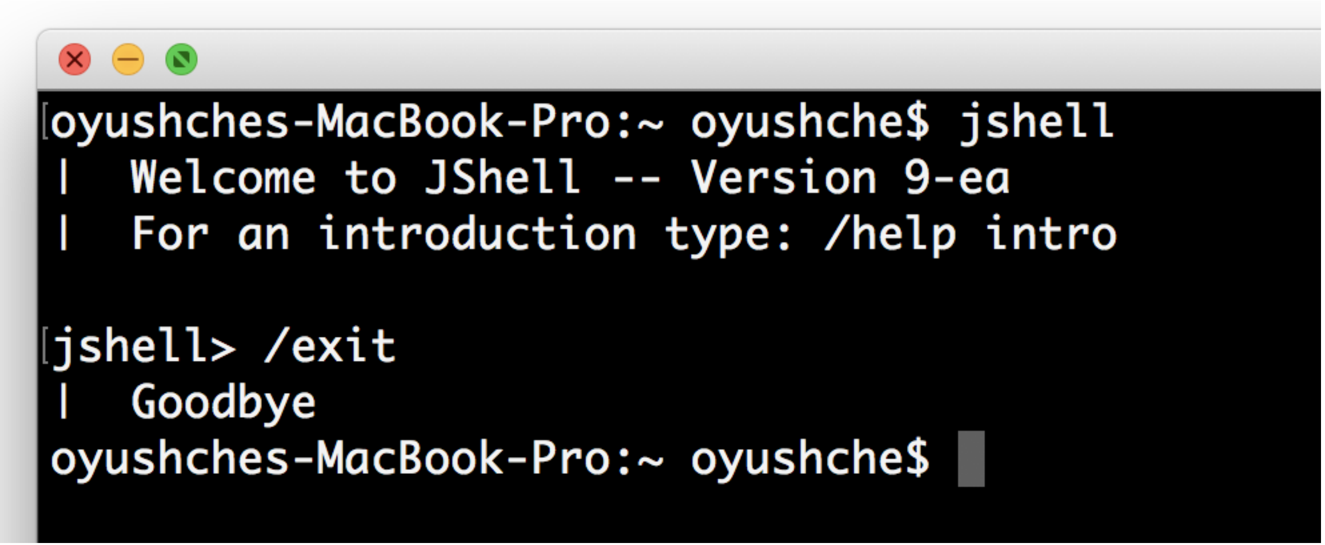
Scala, Ruby, JavaScript, Haskell, Clojure, and Python all have REPLs and all allow small initial programs.

Motivation

JShell adds **REPL** functionality to the Java platform.

System.*out*.println("Hi class") -> now runs !!!

Starting jshell



```
[oyushches-MacBook-Pro:~ oyushche$ jshell
|  Welcome to JShell -- Version 9-ea
|  For an introduction type: /help intro

[jshell> /exit
|  Goodbye
oyushches-MacBook-Pro:~ oyushche$ 
```

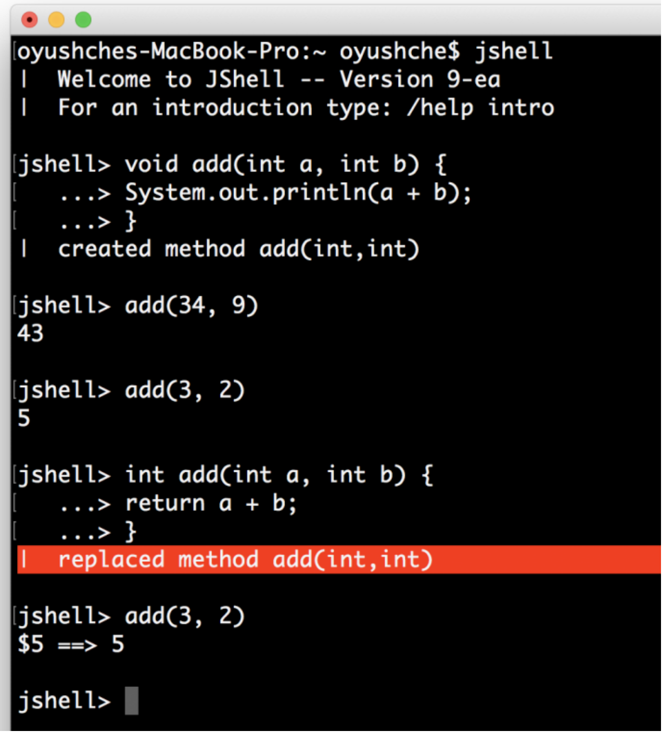
Snippets

JShell accepts:

- expressions
- Java statements
- variable
- method
- class, interface
- imports

Changing definitions

To change the definition simply add it one more time.

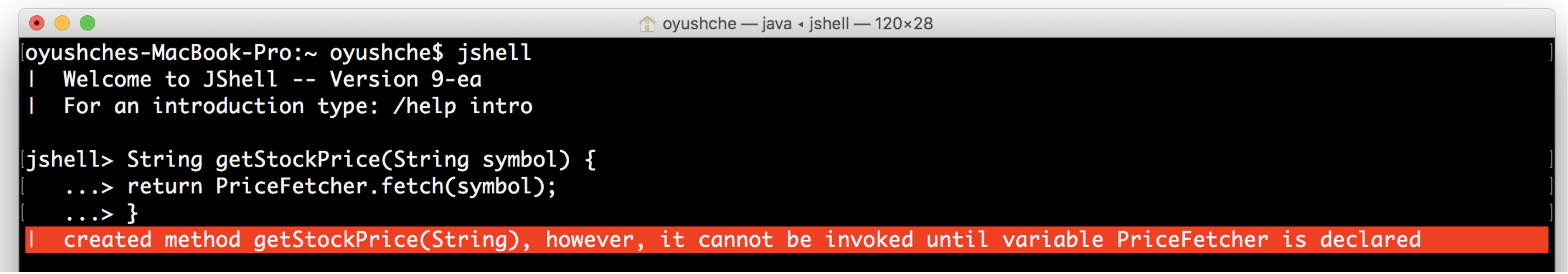


```
oyushches-MacBook-Pro:~ oyushche$ jshell
|  Welcome to JShell -- Version 9-ea
|  For an introduction type: /help intro

jshell> void add(int a, int b) {
   ...> System.out.println(a + b);
   ...> }
|  created method add(int,int)

jshell> add(34, 9)
43

jshell> add(3, 2)
5

jshell> int add(int a, int b) {
   ...> return a + b;
   ...> }
|  replaced method add(int,int)

jshell> add(3, 2)
$5 ==> 5

jshell>
```

Forward Reference

jshel live demo

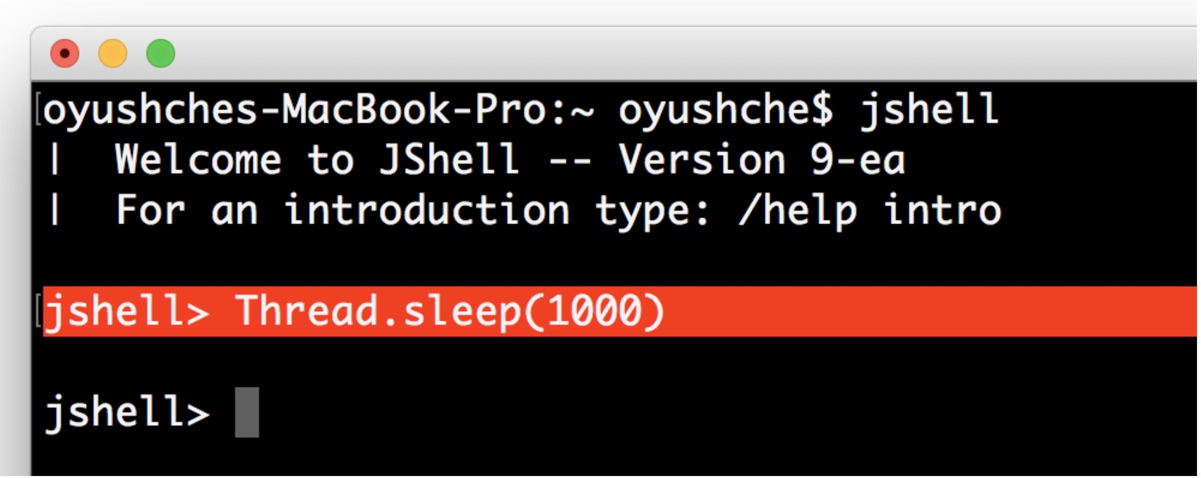
JShell allows you to define methods whose bodies reference methods, variables, or classes which are not yet defined.



```
oyushches-MacBook-Pro:~ oyushche$ jshell
|  Welcome to JShell -- Version 9-ea
|  For an introduction type: /help intro

jshell> String getStockPrice(String symbol) {
   ...> return PriceFetcher.fetch(symbol);
   ...> }
|  created method getStockPrice(String), however, it cannot be invoked until variable PriceFetcher is declared
```

Exceptions

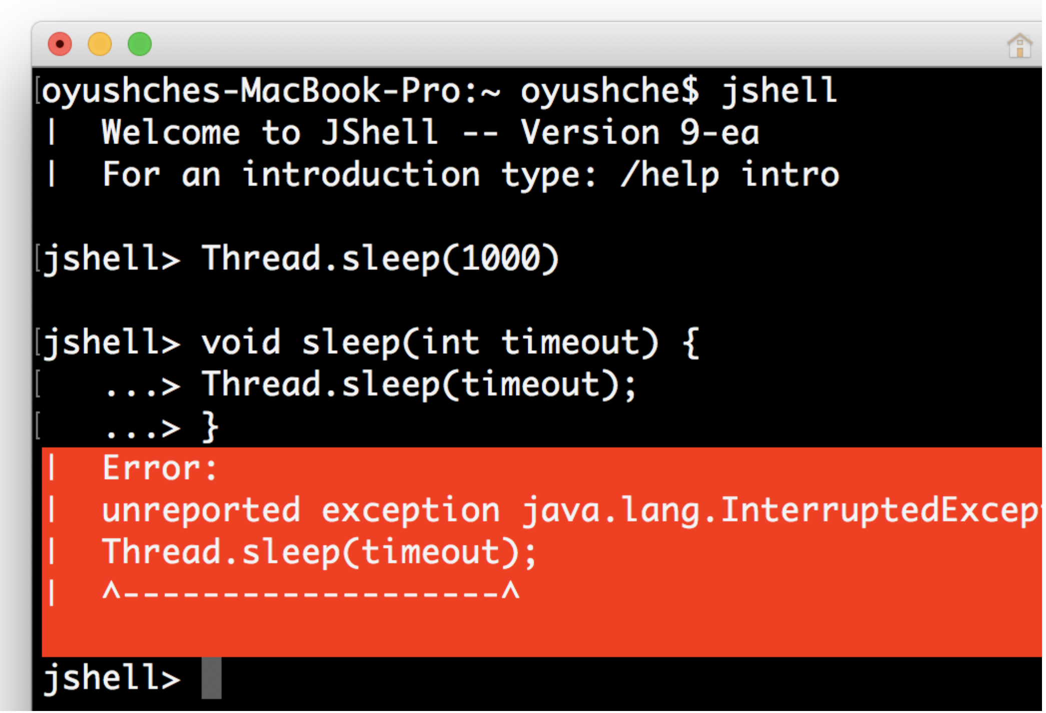
No more try-catch !!!



```
oyushches-MacBook-Pro:~ oyushche$ jshell
|  Welcome to JShell -- Version 9-ea
|  For an introduction type: /help intro

jshell> Thread.sleep(1000)

jshell> 
```

Exceptions

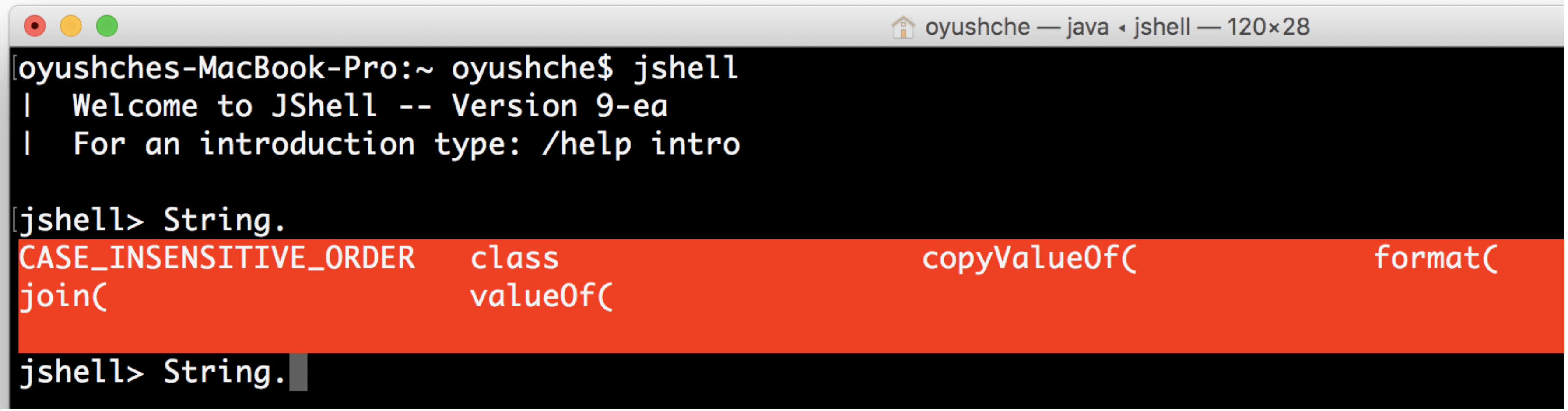
But Not for the methods !!!



```
oyushches-MacBook-Pro:~ oyushche$ jshell
|  Welcome to JShell -- Version 9-ea
|  For an introduction type: /help intro

jshell> Thread.sleep(1000)

jshell> void sleep(int timeout) {
   ...> Thread.sleep(timeout);
   ...> }
|  Error:
|  unreported exception java.lang.InterruptedExcep
|  Thread.sleep(timeout);
|  ^-------------------^

jshell> 
```

Code completion

Use tab for code completion. It works just great!



```
oyushches-MacBook-Pro:~ oyushche$ jshell
|   Welcome to JShell -- Version 9-ea
|   For an introduction type: /help intro

jshell> String.
CASE_INSENSITIVE_ORDER   class                    copyValueOf(            format(
join(                    valueOf(

jshell> String.
```

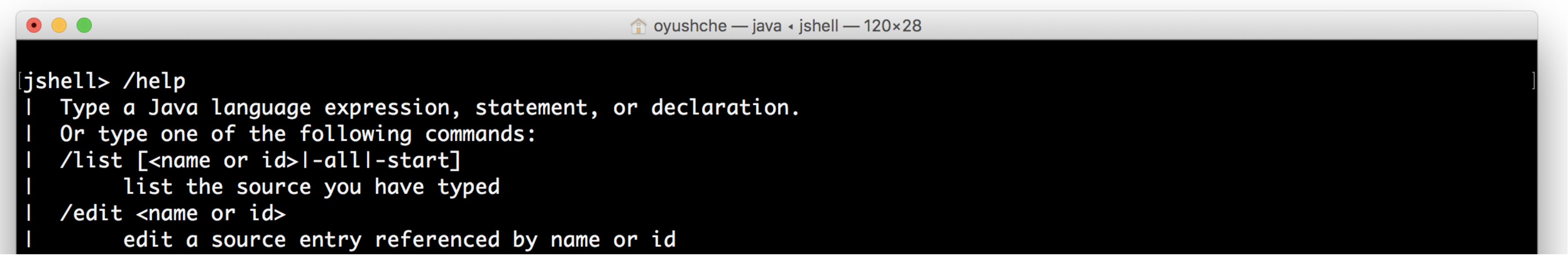
Code transformation

new JFrame <shift+tab> then i - for import

new JFrame() <shift+tab> then v - to create variable

Commands

JShell has a number of commands for controlling the environment and displaying information.



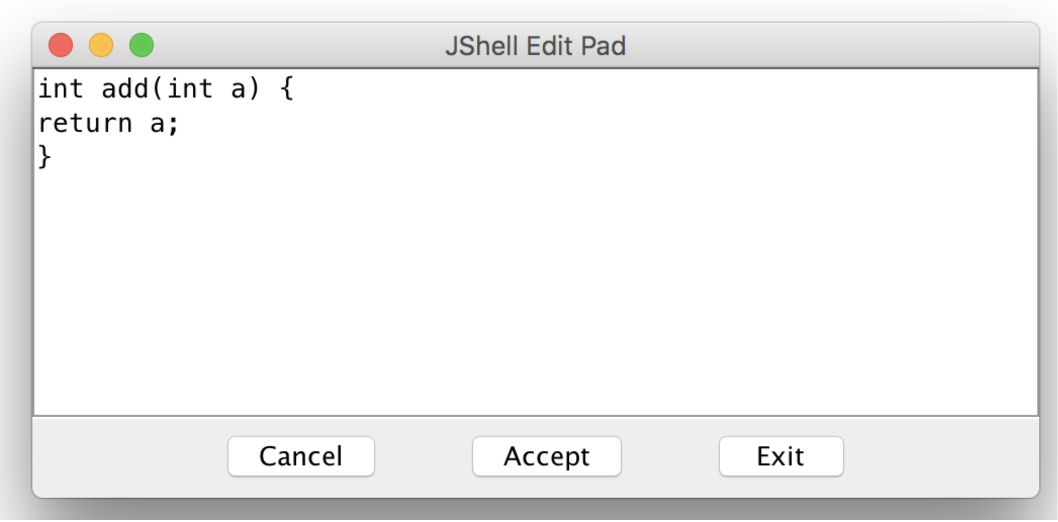
```
jshell> /help
|  Type a Java language expression, statement, or declaration.
|  Or type one of the following commands:
|  /list [<name or id>|-all|-start]
|       list the source you have typed
|  /edit <name or id>
|       edit a source entry referenced by name or id
```

Commands

You can save your snippets to file and then open them when needed.

Commands

/edit - gives you a chance to edit code inside JShell Edit Pad.



```
int add(int a) {
return a;
}
```

Classpath

You can connect external code to be accessed by the jshell tool by setting the classpath.

```
% jshell --class-path myOwnClassPath

jshell> /env --class-path myOwnClassPath
```

Scripts

A JShell script is a sequence of snippets and JShell commands, one per line, in a file.

Scripts

A script can be created externally in an editor, or generated with one of these commands:

```
jshell> /save mysnippets.jsh
jshell> /save -history myhistory.jsh
jshell> /save -start mystartup.jsh
```

Startup scripts

The startup script(s) are loaded each time the jshell tool is reset; reset occurs on initial startup and with resetting commands (**/reset**, **/reload**, and **/env**).

```
jshell> /set start mystartup.jsh
jshell> /reset
```

Task

Create a simple and funny program with jshell, save it to a file and make a demo.