

Exercises: JSNI

- 1.** Make a JavaScript function that generates a random number and puts the result in an alert (dialog) box. Make a GWT button that invokes a JSNI method that calls the JavaScript function.
- 2.** Make a JavaScript function that takes a number as an argument. It then generates a random number, multiplies it times the argument, and pops up the result in an alert box. Make a GWT button that reads a value from a textfield, converts it to a Double (use `Double.parseDouble` with a try/catch block for `NumberFormatException`), and calls a JSNI method that passes that number to the JavaScript function.
- 3.** Make a Java static method that takes a number as an argument. It then generates a random number, multiplies it times the argument, and returns the result. Make a JSNI “native” method that contains JavaScript that calls this Java method. Make a GWT button that reads a value from a textfield, converts it to a Double (use `Double.parseDouble` with a try/catch block for `NumberFormatException`), and passes that number to the JavaScript function. The JavaScript function passes the number to the Java method, gets the result back, and puts the result in an alert box. This is actually the hardest problem in this set of exercises; the actual code to call Java from a JSNI native method is just one line long, but it uses the weird `@yourPackage.client...` syntax. Even worse, if something goes wrong, the error messages are totally unhelpful. If you want, skip this problem and come back and do it at the end if you have time.
- 4.** Make a JavaScript object that represents a person, with `firstName`, `lastName`, and `emailAddress` properties. Make a GWT overlay class for this. Add a `fullName` method that returns the first and last names concatenated together with a space in between. Test based on a JavaScript value.
- 5.** Make an array of those JavaScript objects, and read it into a `JsArray` in Java. Loop down and print out all the first names, last names, and full names.