BlockEntry.java

1   /**
2    * BlockEntry extends SymbolTableEntry by adding a data field to support
3    * nested scoping. The field is a reference to a Symbol table for the block’s
4    * scope.
5    *<p>
6    * BlockEntry is intended to support anonymous inner blocks, in contrast to the
7    * nested scoping blocks defined by a function scope.
8    */
9   public class BlockEntry extends SymbolTableEntry {
10      /**
11       * Construct this with null data fields.
12       */
13       public BlockEntry() {
14         }
15      }
16      /**
17       * Construct this with the given data field value.
18       */
19      public BlockEntry(SymbolTable scope) {
20         super("Anonymous block", null);
21         this.scope = scope;
22      }
23      }
24      /**
25       * Return the string rep of this.
26       */
27      public String toString(int level) {
28         return super.toString(level) + scopeString(level);
29      }
30      }
31      /**
32       * Called by toString to recursively stringify the scope, if non-null.
33       */
34      protected String scopeString(int level) {
35         return scope == null ? "" : 
36         "\n * + indentString(level) + 
37         scope.toString(level);
38      }
39      }
40      /** Local scope for this function */
41      public SymbolTable scope;
42      }