BlockEntry.java

/**
 * BlockEntry extends SymbolTableEntry by adding a data field to support
 * nested scoping. The field is a reference to a Symbol table for the block’s
 * scope.
 *
 * BlockEntry is intended to support anonymous inner blocks, in contrast to <a
 * href= FunctionEntry.html> FunctionEntry </a>, which supports the named
 * scoping blocks defined by a function scope.
 */

public class BlockEntry extends SymbolTableEntry {

    /**
     * Construct this with null data fields.
     */
    public BlockEntry() {
    }

    /**
     * Construct this with the given data field value.
     */
    public BlockEntry(SymbolTable scope) {
        super("Anonymous block", null);
        this.scope = scope;
    }

    /**
     * Return the string rep of this.
     */
    public String toString(int level) {
        return super.toString(level) + scopeString(level);
    }

    /**
     * Called by toString to recursively stringify the scope, if non-null.
     */
    protected String scopeString(int level) {
        return scope == null ? "" : 
            "\n * + indentString(level) + 
          scope.toString(level);
    }

    /** Local scope for this function. */
    public SymbolTable scope;
}