

CLMemberInfo.java

```
1 // Copyright 2013 Bill Campbell, Swami Iyer and Bahar Akbal-Delibas
2
3 package jminusminus;
4
5 import java.io.IOException;
6 import java.util.ArrayList;
7
8 /**
9  * Abstraction of field_info and method_info structures (JVM Spec Section 4.6,
10  * 4.7). CLFieldInfo and CLMethodInfo classes in this file provide concrete
11  * implementations. Instances of these classes are created to store information
12  * about class members (fields and methods) when a class is read using
13  * CLAbsorber or created using CLEmitter.
14  */
15
16 abstract class CLMemberInfo {
17
18     // The fields below represent the members of the field_info and
19     // method_info structures. See JVM Spec Sections 4.6, 4.7 for
20     // details.
21
22     /** member_info.access_flags item. */
23     public int accessFlags;
24
25     /** member_info.name_index item. */
26     public int nameIndex;
27
28     /** member_info.descriptor_index item. */
29     public int descriptorIndex;
30
31     /** member_info.attributes_count item. */
32     public int attributesCount;
33
34     /** member_info.attributes item. */
35     public ArrayList<CLAttributeInfo> attributes;
36
37     /**
38      * Construct a CLMemberInfo object.
39      *
40      * @param accessFlags
41      *         member_info.access_flags item.
42      * @param nameIndex
43      *         member_info.name_index item.
44      * @param descriptorIndex
45      *         member_info.descriptor_index item.
46      * @param attributesCount
47      *         member_info.attributes_count item.
48      * @param attributes
49      *         member_info.attributes item.
50      */
51
52     protected CLMemberInfo(int accessFlags, int nameIndex, int descriptorIndex,
53         int attributesCount, ArrayList<CLAttributeInfo> attributes) {
54         this.accessFlags = accessFlags;
55         this.nameIndex = nameIndex;
56         this.descriptorIndex = descriptorIndex;
57         this.attributesCount = attributesCount;
58         this.attributes = attributes;
59     }
60
61     /**
62      * Write the contents of this class member to the specified output stream.
63      *
64      * @param out
65      *         output stream.
66      */
67 }
```

```

67
68     public void write(CLOutputStream out) throws IOException {
69         out.writeShort(accessFlags);
70         out.writeShort(nameIndex);
71         out.writeShort(descriptorIndex);
72         out.writeShort(attributesCount);
73         for (int i = 0; i < attributes.size(); i++) {
74             CLAttributeInfo attributeInfo = attributes.get(i);
75             attributeInfo.write(out);
76         }
77     }
78
79     /**
80      * Write the contents of this class member to STDOUT in a format similar to
81      * that of javap.
82      *
83      * @param p
84      *         for pretty printing.
85      */
86
87     public void writeToStdOut(PrettyPrinter p) {
88         p.indentRight();
89         p.printf("Access Flags: %s\n", CLFile
90             .fieldAccessFlagsToString(accessFlags));
91         p.printf("Name Index: %d\n", nameIndex);
92         p.printf("Descriptor Index: %d\n", descriptorIndex);
93         p.println();
94         p.printf("// Attributes (%d Items)\n", attributesCount);
95         for (int i = 0; i < attributes.size(); i++) {
96             CLAttributeInfo attributeInfo = attributes.get(i);
97             attributeInfo.writeToStdOut(p);
98         }
99         p.indentLeft();
100     }
101 }
102
103 /**
104  * Representation of field info structure (JVM Spec Section 4.6).
105  */
106
107
108 class CLFieldInfo extends CLMemberInfo {
109
110     /**
111      * Construct a CLFieldInfo object.
112      *
113      * @param accessFlags
114      *         field_info.access_flags item.
115      * @param nameIndex
116      *         field_info.name_index item.
117      * @param descriptorIndex
118      *         field_info.descriptor_index item.
119      * @param attributesCount
120      *         field_info.attributes_count item.
121      * @param attributes
122      *         field_info.attributes item.
123      */
124
125     public CLFieldInfo(int accessFlags, int nameIndex, int descriptorIndex,
126         int attributesCount, ArrayList<CLAttributeInfo> attributes) {
127         super(accessFlags, nameIndex, descriptorIndex, attributesCount,
128             attributes);
129     }
130
131     /**
132      * @inheritDoc
133      */
134
135     public void writeToStdOut(PrettyPrinter p) {

```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```

136         p.printf("Field {\n");
137         super.writeToStdOut(p);
138         p.printf("}\n");
139     }
140
141 }
142
143 /**
144  * Representation of method_info structure (JVM Spec Section 4.7).
145  */
146
147 class CLMethodInfo extends CLMemberInfo {
148
149     /**
150      * Construct a CLMethodInfo object.
151      *
152      * @param accessFlags
153      *         method_info.access_flags item.
154      * @param nameIndex
155      *         method_info.name_index item.
156      * @param descriptorIndex
157      *         method_info.descriptor_index item.
158      * @param attributesCount
159      *         method_info.attributes_count item.
160      * @param attributes
161      *         method_info.attributes item.
162      */
163
164     public CLMethodInfo(int accessFlags, int nameIndex, int descriptorIndex,
165                        int attributesCount, ArrayList<CLAttributeInfo> attributes) {
166         super(accessFlags, nameIndex, descriptorIndex, attributesCount,
167               attributes);
168     }
169
170     /**
171      * @inheritDoc
172      */
173
174     public void writeToStdOut(CPrettyPrinter p) {
175         p.printf("Method {\n");
176         super.writeToStdOut(p);
177         p.printf("}\n");
178     }
179
180 }
181

```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder