CLMemberInfo.java

```java
// Copyright 2013 Bill Campbell, Swami Iyer and Bahar Akbal-Delibas

package jminusminus;

import java.io.IOException;
import java.util.ArrayList;

/**
 * Abstraction of field_info and method_info structures (JVM Spec Section 4.6,
 * 4.7). CLFieldInfo and CLMethodInfo classes in this file provide concrete
 * implementations. Instances of these classes are created to store information
 * about class members (fields and methods) when a class is read using
 * CLAbsorber or created using CLEmitter.
 */

abstract class CLMemberInfo {

    // The fields below represent the members of the field_info and
    // method_info structures. See JVM Spec Sections 4.6, 4.7 for
    // details.

    /** member_info.access_flags item. */
    public int accessFlags;

    /** member_info.name_index item. */
    public int nameIndex;

    /** member_info.descriptor_index item. */
    public int descriptorIndex;

    /** member_info.attributes_count item. */
    public int attributesCount;

    /** member_info.attributes item. */
    public ArrayList<CLAttributeInfo> attributes;

    /**
     * Construct a CLMemberInfo object.
     * 
     * @param accessFlags
     *            member_info.access_flags item.
     * @param nameIndex
     *            member_info.name_index item.
     * @param descriptorIndex
     *            member_info.descriptor_index item.
     * @param attributesCount
     *            member_info.attributes_count item.
     * @param attributes
     *            member_info.attributes item.
     */

    protected CLMemberInfo(int accessFlags, int nameIndex, int descriptorIndex,
            int attributesCount, ArrayList<CLAttributeInfo> attributes) {
        this.accessFlags = accessFlags;
        this.nameIndex = nameIndex;
        this.descriptorIndex = descriptorIndex;
        this.attributesCount = attributesCount;
        this.attributes = attributes;
    }

    /**
     * Write the contents of this class member to the specified output stream.
     * 
     * @param out
     *            output stream.
     */

    public void write(CLOutputStream out) throws IOException {
        out.writeShort(accessFlags);
        out.writeShort(nameIndex);
        out.writeShort(descriptorIndex);
        out.writeShort(attributesCount);
        for (int i = 0; i < attributes.size(); i++) {
            CLAttributeInfo attributeInfo = attributes.get(i);
            attributeInfo.write(out);
        }
    }

    /**
     * Write the contents of this class member to STDOUT in a format similar to
     * that of javap.
     * 
     * @param p
     *            for pretty printing.
     */

    public void writeToStdOut(PrettyPrinter p) {
        p.indentRight();
        p.printf("Acces Flags: %s\n", CLFile
                .fieldAccessFlagsToString(accessFlags));
        p.printf("Name Index: %d\n", nameIndex);
        p.printf("Descriptor Index: %d\n", descriptorIndex);
        p.println();
        p.printf("// Attributes (%d Items)\n", attributesCount);
        for (int i = 0; i < attributes.size(); i++) {
            CLAttributeInfo attributeInfo = attributes.get(i);
            attributeInfo.writeToStdOut(p);
        }
        p.indentLeft();
    }

}

/**
 * Representation of field_info structure (JVM Spec Section 4.6).
 */

class CLFieldInfo extends CLMemberInfo {

    /**
     * Construct a CLFieldInfo object.
     * 
     * @param accessFlags
     *            field_info.access_flags item.
     * @param nameIndex
     *            field_info.name_index item.
     * @param descriptorIndex
     *            field_info.descriptor_index item.
     * @param attributesCount
     *            field_info.attributes_count item.
     * @param attributes
     *            field_info.attributes item.
     */

    public CLFieldInfo(int accessFlags, int nameIndex, int descriptorIndex,
            int attributesCount, ArrayList<CLAttributeInfo> attributes) {
        super(accessFlags, nameIndex, descriptorIndex, attributesCount,
                attributes);
    }

    /**
     * @inheritDoc
     */

    public void writeToStdOut(PrettyPrinter p) {
        p.printf("Field {\n");
        super.writeToStdOut(p);
        p.printf("}\n");
    }

}

/**
 * Representation of method_info structure (JVM Spec Section 4.7).
 */

class CLMethodInfo extends CLMemberInfo {

    /**
     * Construct a CLMethodInfo object.
     * 
     * @param accessFlags
     *            method_info.access_flags item.
     * @param nameIndex
     *            method_info.name_index item.
     * @param descriptorIndex
     *            method_info.descriptor_index item.
     * @param attributesCount
     *            method_info.attributes_count item.
     * @param attributes
     *            method_info.attributes item.
     */

    public CLMethodInfo(int accessFlags, int nameIndex, int descriptorIndex,
            int attributesCount, ArrayList<CLAttributeInfo> attributes) {
        super(accessFlags, nameIndex, descriptorIndex, attributesCount,
                attributes);
    }

    /**
     * @inheritDoc
     */

    public void writeToStdOut(PrettyPrinter p) {
        p.printf("Method {\n");
        super.writeToStdOut(p);
        p.printf("}\n");
    }

}

```