

# Java 클래스파일 비주얼 분석기의 구현

김진광, 박우전

한남대학교 컴퓨터공학과

kimjk@cse.hannam.ac.kr, wjpark@ce.hannam.ac.kr

## A Design and Implementation of Java Class File Visual Analyzer

Jin-Kwang KIM, Woo-Jun PARK

Dept. of Computer Engineering, Hannam University

### 요약

본 논문에서는 자바 컴파일러에 의해 컴파일된 결과인 클래스파일을 효과적으로 분석할 수 있기 위한 비주얼 분석기의 구현에 대해 기술한다. 이 비주얼 분석기는 클래스파일의 모든 세부항목들을 시각적으로 표현하며 원하는 분석 결과를 저장할 수 있게 되어 프로그램의 이해 및 개발을 도와준다. 이 분석기의 구현에서 JDK1.3을 이용하였으며 화면구성은 JBuilder3.0을 사용한다.

### 1. 서론

James Gosling과 Bill Joy의 지휘 하에서 "Oak"라는 코드명으로 고안된 자바는 순수한 객체 지향적 언어로서 간결한 문법과 중간코드 형태인 바이트 코드의 채택으로 플랫폼에 독립적이고 포인터를 제거함으로써 런타임에러의 발생을 감소시키고 실행시간 링크로 인하여 동적이며 멀티스레드를 지원한다. 이러한 특징으로 인해 자바는 최근 몇 년간 급속히 대중화 되어왔다.

본 논문에서는 이러한 특징을 갖는 자바를 통한 프로그램 개발 시에 보다 효율성을 높일 수 있도록 하기 위해 컴파일된 클래스 파일의 상세한 구조를 나타내는 클래스파일 분석기를 설계 및 구현을 하였다.

### 2. 자바 가상기계와 클래스파일

#### 2.1 자바 가상기계

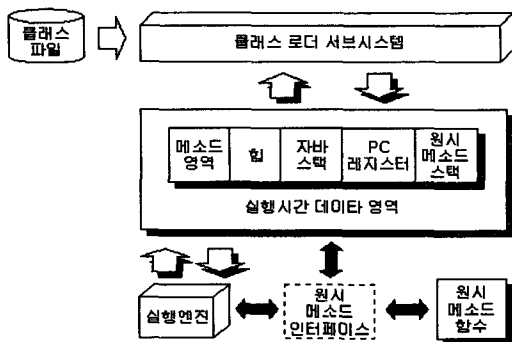


그림1 자바 가상기계의 구조

자바 가상머신은 그림과 같이 클래스 로더 서브시스템(class loader subsystem), 실행엔진(execution engine)과 런타임 데이터 영역(runtime data areas)으로 구성되어있다.

클래스 로더 서브시스템은 클래스나 인터페이스와 같은 형들을 적재(loading)하는 역할을 하며 런타임 데이터 영역은 적재된 클래스 파일로부터 추출된 바이트 코드를 포함한 프로그램의 인스턴스화한 객체, 메소드들의 매개 변수, 리턴값, 지역변수, 계산의 결과 값과 같은 정보를 저장할 기억장소이며 그림과 같이 메소드, 힙, 자바스택, pc 레지스터, 원시 메소드 스택 등으로 구성되어있다. 실행엔진은 자바 가상머신의 심장부로서 적재된 클래스의 메소드들에 포함된 명령어들을 실행하는 역할을 수행한다.

#### 2.2 실행시간 데이터 영역

실행시간 데이터 영역은 다음 바이트 코드의 주소를 담고 있는 PC(Program counter)레지스터와 현재 메소드의 여러 상태들을 저장할 수 있는 지역변수들과 피연산자 스택과 현재 메소드의 실행 환경을 담고 있는 프레임을 저장하는 JVM스택을 가지고 있다. 또한 모든 클래스 인스턴스와 배열에 대한 메모리가 할당되는 데이터 영역인 힙(heap) 영역을 가지고 있고 실행시간 constant pool과 필드와 메소드 데이터 코드 등이 저장되는 메소드영역등으로 구성되어있다.

#### 2.3 클래스파일의 구조

각 클래스파일은 하나의 클래스 혹은 인터페이스의 정의를 담고 있고, 8비트 바이트 스트림으로 구성된다. 클래스파일은 아래의 그림2와 같은 구조로 되어있다.

```

ClassFile {
    u4 magic;
    u2 minor_version;
    u2 major_version;
    u2 constant_pool_count;
    cp_info constant_pool[constant_pool_count];
    u2 access_flags;
    u2 this_class;
    u2 super_class;
    u2 interfaces_count;
    u2 interfaces[interfaces_count];
    u2 fields_count;
    field_info fields[fields_count];
    u2 methods_count;
    method_info methods[methods_count];
    u2 attributes_count;
    attribute_info attributes[attributes_count];
}
    
```

그림2 자바클래스파일의 구조

그림2에서 u2은 unsigned 2byte를 u4는 4byte를 나타내며 처음 4byte인 magic은 magic number로 0XCAFEBABE의 값을 가지며 이 파일이 클래스파일 포맷임을 확인해 준다. minor\_version과 major\_version은 버전 정보를 제공하고 constant\_pool\_count는 상수풀 엔트리의 개수보다 하나 많은 수를 나타내며 constant\_pool은constant\_pool\_count의 수보다 하나 적은 수의 다양한 상수들을 나타내고 있으며 그 형태는 각 엔트리가 가지고 있는 1바이트 tag에 의해 결정된다. this\_class는 현재의 클래스나 인터페이스를 표현하는 상수풀 테이블로의 유효한 인덱스를 가지고 있으며 super\_class는 0혹은 상수풀 테이블로의 유효한 인덱스를 가지고 있다. interface\_count는 이 클래스 혹은 인터페이스의 직접 수퍼인터페이스의 값을 나타내고 이 값만큼의 상수풀 테이블로의 유효한 인덱스를 interface에서 가지고 있다. fields는 fields\_count수만큼 정의된 필드들을 표현하는 구조를 갖는다. method는 method\_count수만큼의 메소드 구조를 갖고 attribute또한 attributes\_count 수만큼의 attribute 구조를 갖는다.

3. 클래스파일 분석기

본 논문에서는 클래스의 다양한 정보를 표현하기 위하여 그림 2에 나타난 클래스의 자료구조 형식대로 화면에 모든 내용을 출력해 주거나 또는 일부분만의 정보만을 출력할 수도 있다. 출력되는 주요한 정보로는 Constant pool과 각 메소드의 해석된 Code 등이 있으며 본 논문에서 구현한 자바 클래스파일 분석기의 전체구성도는 그림 3과 같이 초기화부, 덤프부, 분석부, 이벤트 처리부로 나뉘어져 있다.

비주얼 분석기 초기화부

초기화부는 비주얼 분석기를 초기화하며 화면에 출력할 각 요소를 생성하고 각 요소에 해당하는 이벤트를 연결해주는 부분이다. 화면에 출력할 요소로는 사용자 입력을 위한 메뉴와 메뉴의 각 항목과 대응하는 4개의 버튼이 있고 결과 출력을 위해 그림2의 클래스의 구조와 대응하는 구조를 출력하는 트리와 분석결과를 텍스트로 표시해주는 창이 있다.

16진수 덤프부

덤프부는 비주얼 분석기에 의해 선택된 해당 파일을 16진수로 변환해주는 부분이며 해당 파일을 16진수 표현으로 변환한 것을 가지고 있을 뿐 실제 출력을 하지는 않고 실제 출력은 이벤트 처리부에서 한다.

클래스 파일 분석부

클래스 파일의 분석부는 이 논문의 가장 핵심부분이라고 할 수 있으며 실제 선택된 클래스 파일의 내용을 분석하고 그 내용을 해석하여 가지고 있는 부분이다. 분석내용을 클래스 안에 가지고 있을 뿐 실제 출력하는 작업은 하지 않는고 실제 출력은 이벤트 처리부에서 담당하게 된다.

클래스 파일의 분석 결과는 setComponent 클래스가 가지고 있으며 이 setComponent의 구성은 그림4와 같다.

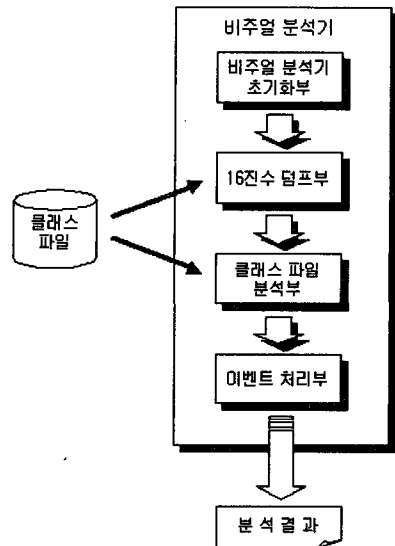


그림 3 비주얼 분석기 전체 구성도

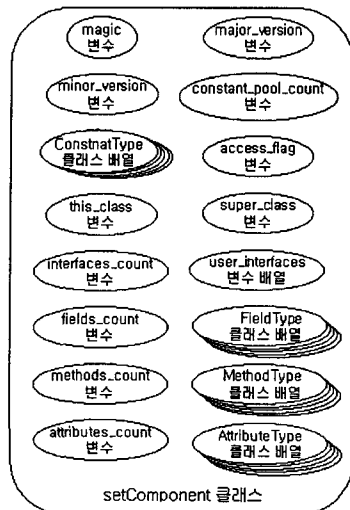


그림 4 setComponent 클래스의 구조  
위의 setComponent클래스는 그림2의 클래스 구조를 가

지고 있으며 특이할 사항은 상수풀의 경우에는 상수풀 배열의 각 내용이 각기 다르기 때문에 ConstantType의 배열이 선언되고 실제 객체는 아래 그림의 ConstantType의 하위 클래스의 인스턴스중 하나가 할당되게 되어있다.

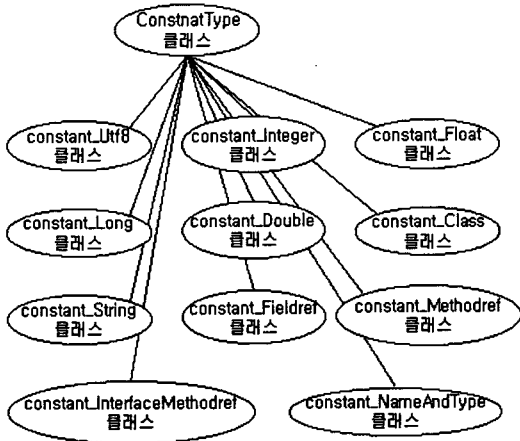


그림 5 ConstantType 클래스의 하위 클래스 구조 분석작업을 할 경우 분석할 해당 파일이 선택이 되면 이 파일에서 각 정보를 분석하여 그 결과를 setComponent 클래스의 각 변수 또는 클래스 인스턴스에 입력을 해놓게 된다. 그리고 출력을 할 경우에는 setComponent의 각 변수 또는 클래스에서 그 결과를 출력하기만 하면 된다.

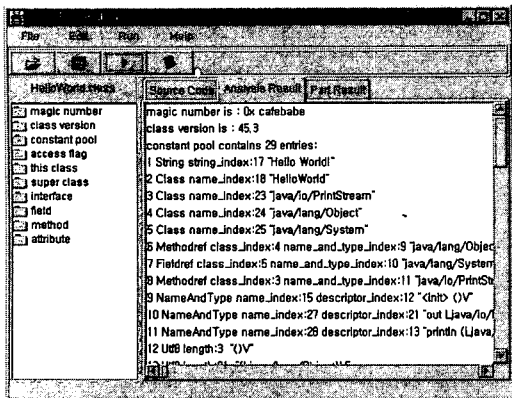
**이벤트 처리부**

이벤트는 비주얼 분석기의 메뉴를 선택하거나 버튼을 누르면 발생되게 된다.

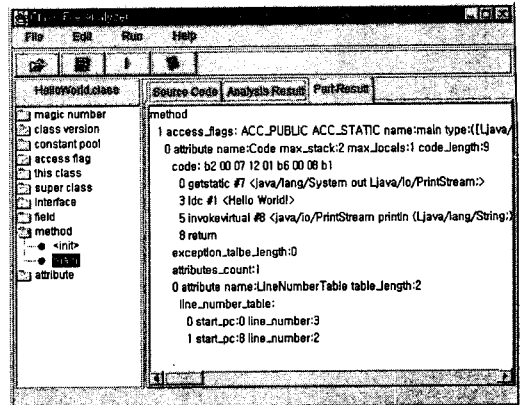
주요한 이벤트로는 open과 analyze가 있으며 open을 하면 클래스 파일을 선택할 수 있는 창이 뜨며 이 클래스 파일을 선택하면 16진수 덤프 내용을 볼 수 있고analyze 버튼을 누르면 현재 선택된 클래스를 클래스 분석부에서 분석을 한후 화면에 출력하게 된다. 또다른 이벤트로써 트리의 각 항목을 마우스로 클릭하게되면 해당하는 항목에 대한 내용이 오른쪽 창에 텍스트로 나타나게 되어있다.

**4. 실행 예**

**1) 분석버튼 실행결과**



**2) 부분분석 실행 결과**



**5. 결론 및 연구 결과의 활용**

본 논문에서 소개한 자바 분석기는 클래스의 구조를 시각적으로 표현함으로써 클래스의 전체적인 흐름과 내용을 쉽게 이해할 수 있게 하였다. 앞으로 본 연구결과를 활용하기 위해서는 현재 구현된 비주얼 분석기 분석결과를 이용하여 Java decompiler와 클래스 내의 메소드 관계 그래프나 클래스 계층구조등의 표시나 실행시간 상태 화면 표시등의 연구들이 뒤따라야 할 것이다.

**참고문헌**

- [1] Cay S. Horstmann and Gary Cornell, core JAVA 1.1 Vol. 1 Fundamentals, Vol. 2 Advanced Features, Sun Microsystems Press ( A Prentice Hall Title), 1997.
- [2] James Gosling, Bill Joy, and Guy Steele, "The Java Language Specification," Addison-Wesley, 1996.
- [3] Mary Campione and Kathy Walrath, "The Java Tutorial, Object-Oriented Programming for the Internet," Addison-Wesley, 1996.
- [4] PETER van der LINDEN, Just Java, SunSoft Press (A Prentice Hall Title), 1996.
- [5] Bjarne Stroustrup, "The C++ Programming Language," 3rd Ed. Addison-Wesley, 1997.
- [6] Arthur van Hoff, Sami Shai, and Orca Starbuck, Hooked on Java, Addison-Wesley, 1996.
- [7] Tim Lindholm and Frank Yellin, The Java Virtual Machine Specification Second Edition, Addison-Wesley, 1997.
- [8] Jon Meyer and Troy Downing, Java Virtual Machine, O'Reilly, 1997.
- [9] <http://java.sun.com/j2se/1.3/>
- [10] 류동항, 정민수 "자바 바이트 코드 분석기의 설계 및 구현", 한국정보과학회 '98 봄학술발표논문집, 제25권 1호, pp 77-79(1998)