

# 자바 클래스 파일에 대한 시각화 브라우저의 설계 및 구현

윤영수<sup>0)</sup> 박상필 고평만  
상지대학교 컴퓨터정보공학부  
{browny3<sup>0)</sup>, in-ppsp, kkman}@mail.sangji.ac.kr

## Design and Implementation of the Visualized Browser for Java Class File

Yoon Young-Soo<sup>0)</sup> Park Sang-Pill Ko Kwang-Man  
School of Computer and Information Engineering, SangJi University

### 요 약

자바 언어는 빠른 속도로 인터넷 및 분산 응용 분야 등에서 활용되고 있으며 단순히 응용 소프트웨어를 개발할 수 있는 프로그래밍 언어 이상으로 활용 범위가 확대되고 있다. 특히, 실행 환경인 자바 가상 기계에 연관되어 다양한 연구가 진행되고 있으며 자바 클래스 파일에 대한 분석 및 응용 분야에 적합한 형태로 정보를 활용할 수 있는 다양한 시도가 진행되고 있다. 본 연구에서도 자바 클래스 파일에 대한 분석을 보다 편리하게 진행하고 클래스 파일이 갖는 정보에 대한 접근을 용이하게 할 수 있는 클래스 파일에 대한 시각화 브라우저를 설계하고 구현하였다. 본 연구에서는 새로운 기술적인 방법론은 없지만 구현된 시각화 브라우저를 활용하여 클래스 파일에 대한 정보 접근 및 분석을 용이하게 할 수 있다.

### 1. 서 론

자바 프로그래밍 언어는 인터넷 및 분산 환경 시스템에서 효과적으로 응용 프로그램을 작성할 수 있도록 설계된 언어로서 객체지향 페러다임 특성 및 다양한 개발 환경을 지원하고 있다. 자바 언어 시스템에서는 응용 프로그램을 플랫폼에 독립적으로 실행시키기 위해 자바 가상 기계에서 클래스 파일을 사용하고 있다. 따라서 클래스 파일은 윈시 프로그램에 대한 모든 정보를 갖고 있으며 이러한 정보를 효과적으로 저장하고 활용할 수 있는 다양한 시도가 많이 진행되어 왔다. 특히, 생성된 클래스 파일 정보에 대한 분석을 통해서 클래스 파일의 최적화와 같은 보다 개선된 실행 방법이 제시되어 왔다[1].

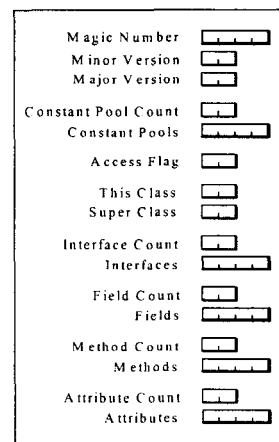
바이너리 형태로 구성된 클래스 파일[3][4]은 자바 가상 기계에서 올바른 수행을 위한 많은 정보를 가지고 있으며 실질적으로 클래스 파일의 구조 및 정보를 추출하기 위한 도구들이 사용되고 있다. 하지만 이러한 도구들은 대부분 텍스트 형식으로 구성되어 클래스 파일의 내용을 접근하는데 많은 불편함을 가지고 있다. 또한 클래스 파일의 핵심 부분인 바이트코드를 이해하기 위해서는 복잡한 클래스 파일에 대한 접근이 제한되고 있다. 따라서 본 논문에서는 클래스 파일의 정보를 보다 편리하게 접근할 수 있도록 클래스 파일의 구조를 6개 핵심 부분으로 나누어 재구성하였으며 각각의 정보를 시각화하여 표현하였다. 이러한 클래스 파일에 대한 시각화 브라우저는 사용자에게 클래스 파일의 접근을 보다 편리하게 할 수 있으므로 자바 윈시 프로그램의 보다 빠른 이해 및 클래스 파일을 활용하는 다양한 응용 분야에 실질적인 도움을 제공할 수 있다.

본 논문의 구성은 제 2장에서 자바 컴파일러에 의해 생성되는 클래스 파일의 구조를 간략히 설명하여 기존의 클래스 파일 분석 도구에 대해 고찰한다. 제 3장에서는 본 논문에서 설계하고 구현한 시각화 브라우저에 대한 구현 과정 및 실험 결과를 기술하였다. 마지막으로 제 4장에서는 결론 및 향후 연구 내용에 대해 기술하였다.

### 2. 기반 연구

#### 2.1 클래스 파일

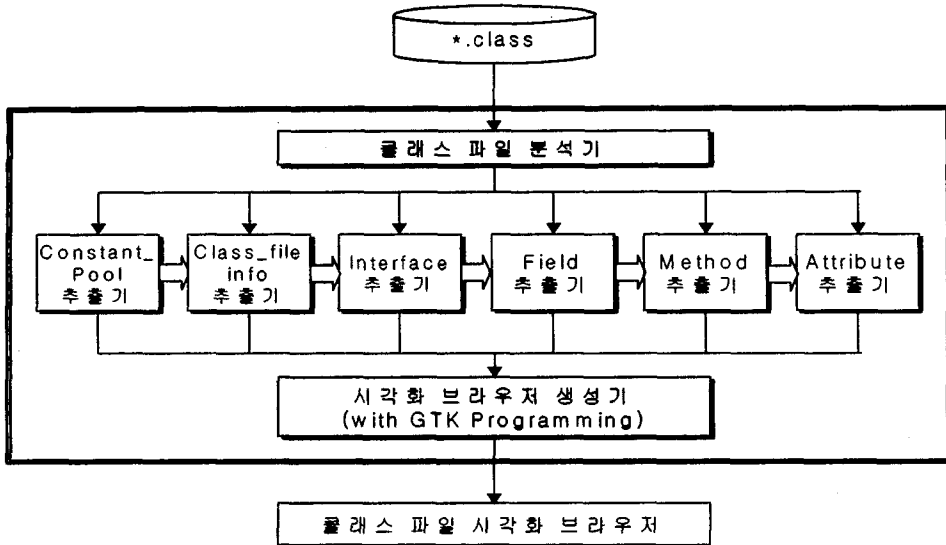
자바 클래스 파일은 플랫폼 독립성을 위해 설계된 바이너리 코드인 8 비트의 이진 스트림으로 구성되며 데이터그램은 "Big-endian"의 순서로 저장되며 [그림 1]과 같은 구조를 가지고 있다[4].



[그림 1] 클래스 파일 구조

이 논문은 한국과학재단의 특정기초연구(과제번호: R01-2002-000-00041-0)지원에 의한 것임.

클래스 파일에서 상수 정보는 constant\_pool[]에 저장되며 상수의 자료형에 따라 읽어오는 크기가 다르다.



[그림 2] 전체 시스템 구성도

methods[]에는 클래스 및 인터페이스에 포함된 메소드 정보를 저장하며 핵심 부분인 니모닉 코드를 저장하고 있다. attributes[]에는 소스 파일 이름, 디버깅 정보 등과 같은 클래스 파일의 각종 부가 정보가 저장된다.

## 2.2 클래스 파일 분석 도구

선 마이크로 시스템즈에서 제공되는 JDK에서는 javap -c 옵션을 이용하여 클래스 파일의 내부 구조를 접근할 수 있다. 클래스 파일의 정보는 메소드 부분에 대한 Opcode 부분으로서 JVM에서 수행되는 역할을 상세하게 볼 수 있다.

DJ Java Decompiler는 시각화 브라우저를 사용해 클래스 파일의 정보와 역컴파일까지 가능하게 한 도구이다. jad를 이용하여 클래스 파일을 분석하였기 때문에 JVM에서 지원하는 javap의 기능과 유사하여 클래스 파일의 내용 전체를 보여주지 못하고 JVM에서 실행되는 메소드를 중심으로 Opcode에 대한 정보를 보여준다[5].

DumpClass는 Oolong의 Dumpclass로서 본 연구 결과와 같이 클래스 파일에 대한 모든 정보를 알 수 있는 도구이지만 텍스트 형태로 표시되기 때문에 일반 사용자가 클래스 파일에 대한 구체적인 지식 없이 접근하기 쉽지 않은 도구이다[4].

ClassCracker는 Mayon Software Research의 클래스 파일 분석 도구로서 java로 작성되어 클래스 파일을 덤프시켜 내용을 볼 수 있도록 한 도구이다. 본 논문에서 제시하는 시각화 브라우저와 유사한 점이 있으나 전체 클래스 파일에 대한 내용을 텍스트 형식과 같이 동시에 표현하므로써 클래스 파일에 대한 접근이 쉽지 않다[6].

## 3. 시각화 브라우저의 설계 및 구현

### 3.1 개요

자바 클래스 파일은 자바 원시 프로그램에 대한 8-비트

형식의 바이트 단위로 구성된 바이너리 데이터를 의미한다. 이러한 바이너리 형식의 데이터는 각각의 지정된 의미에 의해 자바 가상 기계에서 수행될 수 있는 형태로 번역된다. 응용 프로그래머의 목적에 따라 이러한 클래스 파일의 형태를 분석한 후 각각의 정보를 접근하거나 필요에 따라 클래스 파일에 대한 분석하는 경우에는 보다 용이하게 접근하는데 한계를 가지고 있다. 본 논문에서는 자바 클래스 파일에 대한 분석을 보다 편리하게 진행하고 클래스 파일이 갖는 정보에 대한 접근을 용이하게 할 수 있는 클래스 파일에 대한 시각화 브라우저를 설계하고 구현하였다. 시각화 브라우저는 [그림 2]와 같이 자바 컴파일러(javac)에 의해 생성된 클래스 파일을 입력을 받아 클래스 파일의 핵심 요소를 6개의 부분으로 분리한 후 각각의 정보를 사용자가 보다 편리하게 접근할 수 있도록 시각화된 형태로 재구성하였다. 또한 각 정보간의 관계를 유기적으로 표현하여 기존의 텍스트 형식에 비해 클래스를 구성하는 정보간의 관계를 쉽게 접근할 수 있도록 구현하였다.

### 3.2 설계 및 구현

클래스 파일 브라우저는 리눅스 환경에서 GIMP 개발을 위한 도구로 개발된 GTK(GIMP Tool Kit)[7]을 사용하여 인터페이스를 구성하였다. 클래스 파일 분석기는 \*.class 파일을 입력으로 받아 순차적으로 Constant\_Pool, Class\_file\_information, Interface, Fields, Methods, Attributes와 같이 총 6개의 부분으로 분리하여 각각에 대한 정보를 GTK를 이용하여 출력한다.

Constant\_Pool 부분은 magic\_number, major\_version, minor\_version, Constant\_pool\_count 필드로 구성되어 있다. magic\_number 필드에서는 입력된 파일이 올바른 클래스 파일인지는 판별하며 Constant\_pool\_count 필드에

서는 Constant\_pool의 개수 및 내용을 표시한다.

Class\_information 부분은 클래스의 접근 플래그를 나타내는 accessflag, 클래스 자신의 이름을 나타내는 this\_class 및 슈퍼 클래스의 이름을 나타내는 super\_class 필드로 구성되어 있다.

Interface 부분에서는 인터페이스의 개수 및 실질적인 내용을 interface\_count 필드와 interface 필드에서 각각 표시한다.

Fields 부분에서는 클래스 변수 또는 인스턴스 변수들의 개수를 fields\_count 필드에서 표시하며 Field access flag 포함한 실질적인 플래그의 내용은 fields에서 나타난다.

Methods 부분에서는 메소드 및 인스턴스 메소드의 개수를 methods\_count에 나타내며 메소드의 따른 속성들을 포함하여 실질적인 메소드 내용이 표시된다.

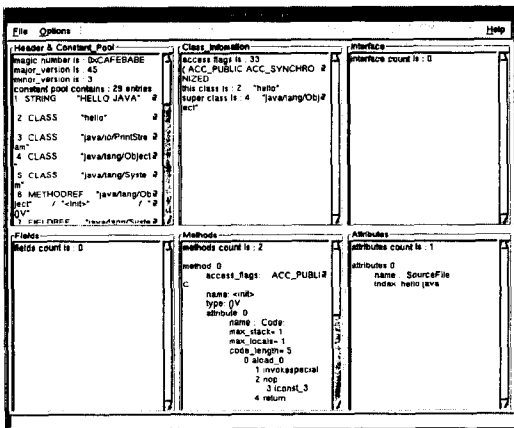
Attributes 부분에서는 실질적인 속성의 개수 및 내용이 나타난다.

### 3.3 실험 및 분석

자바 언어로 작성된 프로그램에 대해 현재까지 완성된 클래스 파일에 대한 시각화 브라우저의 결과는 [그림 3]과 같다.

```

1 public class hello {
2     public static void main(String args[]) {
3         System.out.println("HELLO JAVA");
4     }
5 }
    
```



[그림 3] 클래스 파일에 대한 시각화 브라우저

자바 언어에 대한 클래스 파일을 읽어들이 클래스 파일의 구조를 분석한 후 6개 부분으로 구성하여 출력한다. 각 창은 보다 자세한 정보 접근을 위해 다시 분리하여 볼 수 있으며 복잡한 클래스 파일의 구조에 대해서도 많은 정보를 보다 편리하게 접근할 수 있다. 현재까지 완성된 결과를 이용하여 시각화 브라우저에서 클래스 파일

에 대한 변형을 할 경우 변경된 내용을 다시 원시 프로그램 및 원본 클래스 파일에 반영할 수 있도록 보완 연구중이다.

### 4. 결론 및 향후 연구

바이너리 형태로 구성된 클래스 파일은 자바 가상 기계에서 올바른 수행을 위한 많은 정보를 가지고 있으며 실질적으로 클래스 파일의 구조 및 정보를 추출하기 위한 도구들이 사용되고 있다. 하지만 이러한 도구들은 대부분 텍스트 형식으로 구성되어 클래스 파일의 내용을 접근하는데 많은 불편함을 가지고 있다. 또한 클래스 파일의 핵심 부분인 바이트코드를 이해하기 위해서는 복잡한 클래스 파일에 대한 접근이 제한되고 있다. 따라서 본 논문에서는 클래스 파일의 정보를 보다 편리하게 접근할 수 있도록 클래스 파일의 구조를 6개 핵심 부분으로 나누어 재구성하였으며 각각의 정보를 시각화하여 표현하였다. 이러한 클래스 파일에 대한 시각화 브라우저는 사용자에게 클래스 파일의 접근을 보다 편리하게 할 수 있으므로 자바 원시 프로그램의 보다 빠른 이해 및 클래스 파일을 활용하는 다양한 응용 분야에 실질적인 도움을 제공할 수 있다. 현재까지 완성된 결과는 클래스 파일을 6개의 부분으로 나누어 각각의 정보를 보다 편리하게 접근할 수 있도록 표현하였으며 각각의 정보를 보다 시각적(계층적 구조)으로 표현하는데 보완 연구를 진행중이다. 또한 사용자 시각화 브라우저를 통해서 클래스 파일을 변경할 경우 변경된 정보를 반영하여 클래스 파일을 재구성하고 재구성된 정보를 다시 소스 프로그램에 반영할 수 있는 연구를 진행중이다.

### 5. 참고 문헌

- [1] Jon Meyer and Troy Downing, JAVA Virtual Machine, O'REYLLY, 1997.
- [2] Ken Arnold and James Gosling, The Java Programming Language, Sun Microsystems, 1996.
- [3] Tim Lindholm, Frank Yellin, The Java Virtual Machine Specification, O'REYLLY, 1999.
- [4] Joshua Engel, Java Virtual Machine Programming, Info-book, 2000.
- [5] <http://members.fortunecity.com>
- [6] <http://www.tip.net.au>
- [7] <http://www.gtk.org/tutorial>