

네트워크 컴퓨팅 솔루션

金 泰 賢

韓國 썬.마이크로시스템즈(株)

월드-와이드-웹(WWW)은 전세계의 수많은 컴퓨터가 연결된 온라인의 세계이다. 이러한 WWW 안에는 수많은 정보들이 저장되어 있고, 그러한 정보들은 네트워크를 통하여 공유되고 있다. 이러한 인터넷의 세계에서 더 이상 문자만으로는 정보시스템을 구성할 수 없게 되었다. 왜냐하면, 웹의 사용자들은 정보들을 보고 선택하는 기회를 가지고 있고, 하이퍼텍스트로 구성된 정보보다는 그래픽, 문자, 동영상 및 음성으로 구성된 정보를 선택하는 추세이기 때문이다. 즉, 사용자와 응용프로그램간에 실시간의 작동 및 시청각적인 상호 작동성을 선호하고 있기 때문이다.

이러한 이유로 웹은 빠른 속도로 발전하여 왔다. 하지만, 실시간 처리, 동영상, 다양한 프로토콜의 지원 등을 하지 못하는 문제점들이 나타나게 되었다. 이러한 문제점을 해결하여 주는 새로운 해결책이 등장하였고, 그것이 자바(Java) 언어이다. 자바는 기존의 웹에서 결여된 상호 작동성, 실시간 처리 능력, 동화상등을 제공하고 있기 때문에, 자바를 이해하는 웹 브라우저(Web Browser)를 통하여 상호 작동적이고 시청각적인 응용프로그램을 사용할 수 있다.

I. 자바란 무엇인가?

그러면, 자바는 무엇일까? 자바는 네트워크를 통하여 전달되어질 수 있는 실행 가능한 내용을 만들 수 있도록 썬.마이크로시스템에서 개발한 프로그래밍 언어이다. 다시 말하면, 자바 언어를 사용하여 실행 가능한 내용을 만들 수 있도록 구성된 소프트웨어 도구들의 모음을 의미한다. 이러한 도구중의 하나가 자바 언어로 만들어진 자바 생성 코드(Java generated Code)를 실행할 수 있는 브라우저인 핫자바(HotJava)이다. 핫자바는 자바의 실행 가능한 내용을 출력할 수 있는 번역기의 기능을

가지고 있는 가장 최신의 웹 브라우저이다.

여기서, 실행 가능한 내용은 자바를 사용할 수 있는 브라우저와 사용할 수 없는 브라우저를 구분할 수 있는 척도라고 할 수 있다. 자바를 이해할 수 없는 브라우저에서는, 가져온 내용이 멀티미디어 서류 형식의 하나인 MIME(Multipurpose Internet Mail Extensions)으로 구성되어 있다. 내용의 이러한 서술내용은 단지 브라우저 혹은 이미지, 음성과 같은 멀티미디어를 출력하기 위한 보조 응용프로그램에서 출력되어질 수 있다. 물론, 자바를 이해하는 브라우저에서는 MIME 서술 내용에 의해 정의된 내용을 가져올 수도 있고 출력할 수도 있다. 하지만, 중요한 차이는 <CODE>APPLET <CODE>로 정의된 특별한 하이퍼텍스트를 이해한다는 것이다. APPLET을 포함한 웹 페이지를 가져올 때, 자바를 이해하는 브라우저는 그 페이지에 관련된 애플릿(Applet)이라고 불리는 자바 프로그램의 특별한 종류를 이해하고, 브라우저는 그 애플릿의 실행에 대하여 서술된 정보를 가진 파일을 가져오게 된다. 이 정보를 가진 파일은 자바 바이트 코드(bytecode)라고 불리는 형태로 쓰여져 있고, 브라우저는 이러한 자바 바이트 코드를 번역하고 사용자의 시스템 상에 프로그램을 실행하게 된다. 이러한 일련의 절차와 수행은 사용자의 특별한 요청 없이 자동적으로 발생하게 된다. 이러한 이유 때문에 자바를 이해하는 브라우저를 사용하는 사람들은 기존의 웹에서 사용하는 하이퍼텍스트의 정보만이 아니라 네트워크상에서 이미지, 음성 및 동영상을 가진 응용프로그램을 사용할 수 있게 되는 것이다.

이러한 자바는 자바의 아버지로 불리는 썬.마이크로시스템즈의 James Gosling에 의하여 만들어진 이후, 1995년 7월에 SunWorld online에 하나의 논설로 등장하여 이 세상에 나타나게 되었다. 그는 고객을 위한 프로그램들이 특정의 CPU에서만 작동하던 것을 하나의 프로그램을 만들면 모든 CPU에서 작동될 있도록 개발하려고 하였다.

이러한 목적을 위하여 그는 처음에 객체 지향적(Object-Oriented) 언어인 C++에 초점을 맞추었다. 하지만, C++로는 상이한 장치를 가지고 통신

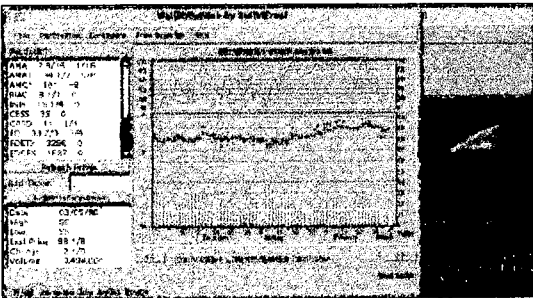
하는 분산 네트워크를 지원하도록 시스템을 만들기 위한 조건을 충족시킬 수 없어서, C++를 버리고 새로운 언어를 만들기로 하였고 그 언어가 바로 자바이다. 자바는 1995년 유명한 웹 브라우저 생산 업체인 Netscape사에서 자바를 이해할 수 있는 새로운 버전의 넷스케이프 브라우저 2.0을 만들게 되어 새로운 웹의 해결책 및 프로그래밍 언어로 자리를 잡아가고 있고 대부분의 컴퓨터 생산 업체 및 소프트웨어 생산 업체들이 라이선스를 받아 자바로 만든 프로그램들을 출시할 예정으로 있기 때문에, 상업적인 가치도 높다고 하겠다.

자바는 프로그래밍 언어로 프로그래머들 사이에서 잠재성을 인정받고 있고, 많은 응용 프로그램들이 나오고 있다. 먼저, 썬에서는 자바 언어에 대한 개발 도구들의 모음인 자바 개발자 도구(Java Developer's Kits)을 자바 홈 페이지를 통하여 공개하고 있다. 또한 자바를 이해하는 웹 브라우저인 핫자바 및 홈페이지 제작 및 개발을 위한 도구인 자바 워크샵을 발표하였고, 자바 언어로 만든 프로그램과 관계형 데이터베이스와의 연결을 위한 JDBC와 네트워크 및 시스템 관리를 위한 소프트웨어들을 자바 언어로 만든 솔스티스 제품군을 출시하였다. 그리고, 넷스케이프와 같은 소프트웨어 공급업체들도 상업용 소프트웨어를 계속하여 출시하고 있다. 현재 이러한 소프트웨어들은 약 40개 정도가 된다. 이 안에는 개발 환경을 지원하는 각 시스템들을 위한 컴파일러 및 Drag&Drop 도구들, 데이터베이스와의 연계를 도와주는 프로그램들, 웹 브라우저들, 상업용 프로그램들이 있다. 올해 안으로 현재 사용하고 있는 많은 프로그램들 중에서 상당수가 자바 언어로 만들어질 예정이므로, 조만간 자바 언어의 전성기가 도래할 것이다.

또한, 자바 언어로 만들어지는 칩을 국내의 LG에서 만들기로 발표되었고, 미래형 네트워크 컴퓨터의 운영체계가 자바를 지원하는 코나(Kona)로 썬에서 발표되어, 자바 언어의 폭 넓은 활용이 기대되고 있다.

II. 자바의 특성

자바 언어의 가장 큰 특성은 동영상(Animation)이다. 자바 언어로 만화, 이미지, 그래픽 혹은 특정 글씨의 움직임은 자유자재로 구성할 수 있다. 예를 들면, 우리 주변에서 흔히 볼 수 있는 증권사를 방문하면 한쪽 벽 가득히 주식시세를 나타내는 전광판과 각종 데이터와 그래프를 제공하는 단말기가 있다. 이제 우리는 웹 상에서 운영되는 자바 언어로 만들어진 프로그램을 통하여, 증권사에 방문을 하지 않더라도 집에서 혹은 사무실에서 주식시세 변동 및 그래프를 알아 볼 수 있다. 현재, 미국의 월 스트리트 저널의 홈 페이지에서는 자바 언어로 만든 프로그램을 통하여 유료의 서비스를 이미 제공하고 있다.



(월스트리트 홈페이지의 한화면)

다음의 특성은 상호작용(Interactivity)을 통한 프로그램을 작성할 수 있다는 것이다. 이것은 상당히 주목할만한 특성이다. 왜냐하면, 게임 혹은 대화형 프로그램을 만들 수 있다는 것이다. 이미 웹 상에서 테트리스 게임을 즐길 수 있고, 앞으로 웹 상에서 네트워크를 통하여 여러 명의 사용자가 동시에 게임을 즐길 수 있게 될 것이다. 또한, 상호 작동적인 산술기능을 제공하고 있다. 우리는 일상 업무에서 스프레드시트를 많이 사용하고 있다. 이러한 프로그램도 자바 언어로 만들게 되면 손쉽게 웹 상에서 운영이 가능한 것이다. 현재 그룹웨어로 가장 많이 사용되고 있는 로터스 노츠가 자바 버전으

로 출시될 예정이다.

또한 분산 네트워크 상에서의 프로그램 작성이 용이하고, 대화방과 같이 서로 대화를 나누는 통신 프로그램의 작성도 용이하다. 이제 자바를 통하여 프로그래머들은 그들의 프로그램을 하나의 시스템에서 개발을 하면 상이한 시스템에서도 운영할 수 있을 뿐만 아니라, 손쉽게 동영상을 처리하고 또한 분산처리 와 상호 작동성을 얻을 수 있는 기회를 가지게 된 것이고, 그러한 응용프로그램의 사용자들은 좀더 안정적이며 상호 작동적인 프로그램을 사용할 수 있게된 것이다.

III. 자바 프로그램과 HTML

자바 프로그래밍 언어는 네트워크를 통하여 실행 가능한 내용을 전달하는데 주목적이 있다. 이러한 자바를 이해하기 위해서는 먼저 어떻게 자바언어가 웹의 하이퍼 텍스트를 사용하여 상호 작동하는가를 알아야 한다. 이것을 위해서는 자바 프로그램과 연결된 웹의 HTML을 통하여 웹상에서 applet을 불러오고, 자바를 이해하는 브라우저 상에서 자바 applet을 가진 페이지가 상호 작동하는지를 살펴보면 된다. 웹 상에서 정보를 전달하는 언어인 자바는 APPLET이라고 불리는 특정한 표식을 사용한 웹의 HTML에 연결이 되고, 이러한 APPLET이라는 특정의 표식을 통해 applet의 자바 바이트 코드를 확인한다. 확인이 끝나면 사용자의 브라우저가 자바를 이해할 수 있는지를 검색한 후 applet은 사용자의 브라우저에서 운용된다.

HTML로 작성된 APPLET 표식은 웹 페이지에 포함된 applet이라 불리는 자바 프로그램의 이름을 정의하고 있다. 그 applet의 이름은 class name이라고 한다. 이 이름은 applet을 실행하는 수행 가능한 bytecode와 연계되어있다. 예를 들어, 다음의 HTML예제를 보면 어떻게 웹 문서에 applet을 포함시킬 수 있는지 이해하게 될 것이다.

```

<HTML>
<HEAD>
  <TITLE>First Hello Applet</TITLE>
</HEAD>
<BODY>
  <P> "This is it!" <APPLET code=
    "HelloWorld"> </APPLET>
</BODY>
</HTML>

```

물론, 위에 지정된 code name에 해당되는 HelloWorld라고 불리는 applet을 위한 자바 소스 코드를 만들어야만 한다. 그것을 다음과 같이 만들어 보자.

```

import java.awt.Graphics;
public class HelloWorld extends java.applet.
Applet{
public void init(){
  resize(150,25);
}
public void paint(Graphics g){
  g.drawString("Hello world!", 50, 25);
}
}

```

이렇게 만든 자바 소스 코드는 java라는 확장자를 가지고 있어야만 한다. 즉, 파일의 이름을 HelloWorld.java라는 파일로 저장을 한 후, 다음과 같이 자바 컴파일러를 이용하여 자바 bytecode를 만들어야 한다.

```
$ javac HelloWorld.java
```

만약, 여기서 어떠한 오류도 없었다면 class라는 확장자를 가진 자바 bytecode가 classes라고 불리는 디렉토리 안에 HelloWorld.class처럼 만들어진 것이다. 이 상태에서 보면, HelloWorld.html파일, HelloWorld.java파일과 HelloWorld.class파일을 가지게 된다.

자바는 자바를 이해하는 웹 브라우저를 가지고

연계하여 사용할 수 있는 객체 지향적 프로그래밍 언어이다. 이 브라우저는 자바 언어 컴파일러에 의하여 만들어진 자바 bytecode를 해석할 수 있다. 자바의 기술적인 구조는 시스템에 독립적이라는 것이다. 지금까지와는 다르게, 프로그래머들은 사용자의 컴퓨터의 기종에 상관없이 자바 프로그램을 만들 수 있다. 대신에, 자바를 이해하는 핫자바와 같은 브라우저가 사용자의 시스템에 맞게 조정하는 것이다. 이것이 자바의 기술적인 구조의 주요 특성이다. 이것은 자바 언어가 시스템에 종속적인 실행 파일을 만드는 것이 아니라, Lisp 언어처럼 번역 가능한 코드를 생성하기 때문에 가능한 것이다.

IV. 프로그래밍 언어로서의 자바 언어

▶ 자바는 단순하다.

C++에 기초를 두고 거의 사용하지 않는 부분과 많이 사용하는 부분의 상당부분을 제거하여 더욱 객체 지향적인 모습을 가지고 있다. C++는 매우 강력한 객체 지향적 프로그래밍 언어이다. 하지만, 강력한 다른 언어와 마찬가지로 코드를 만들어 있어서 프로그래머들을 어렵게 만든다. 특히, 코드의 유지보수에 많은 시간과 노력을 빼앗기게 된다. 자바는 이러한 측면을 제거하여 단순하게 만들어 졌다.

▶ 자바는 객체 지향적이다.

C++처럼 자바는 소프트웨어를 만들기 위해 객체 지향적인 접근을 할 수 있도록 한다. 기존의 객체 지향적인 언어와 마찬가지로 객체, 방법, 전승과 데이터의 요약 등의 객체 지향적 모습을 지원하고 있다.

▶ 자바는 분산적이다.

C 혹은 C++과 다르게, 자바는 특별히 네트워크 환경에서 작업할 수 있도록 구성되어 있다. 자바는 HTTP 혹은 FTP등과 같은 프로토콜을 포함하여

인터넷 상에서 사용되는 TCP/IP를 사용하는 통신을 위한 class의 대형 라이브러리를 제공하고 있다. 자바코드는 프로그래머들이 C 혹은 C++를 사용하기 위해 자신의 시스템을 접속하는 것과 마찬가지로 웹의 URL을 통하여 자원들을 조작할 수 있게 한다. 이러한 이유로 분산처리를 위한 프로그래밍이 손쉬워진다.

▶ 자바는 자바 바이트 코드를 번역하는 기능을 가졌다.

이 때문에 하나의 자바 프로그램을 자바번역기를 가진 상이한 시스템 상에서 운영할 수 있는 것이다. 즉, 기존의 프로그래밍 언어들이 시스템에 종속적인 실행 파일을 만드는 것과는 다르게 단지 번역 가능한 자바 바이트 코드를 만들어 번역기를 통해 번역하는 방법을 사용함으로써 상이한 시스템에서의 운영이 가능한 것이다.

▶ 자바는 강인하다.

기존의 언어로 구성된 프로그램들은 그것이 가지고 있는 문제 때문에 손쉽게 파괴될 수 있다. 강인한 언어는 프로그래머들의 코드에 대하여 엄격한 검사를 한다. 이런 측면은 기존의 C보다는 C++에서 좀더 강화되었다. 하지만, C에서의 잔재를 완전히 버리지는 못하였다. 자바는 포인터를 제거하여 이러한 잔재들을 버리고 가장 강인한 언어로 만들어졌다.

▶ 자바는 보안적이다.

자바는 네트워크 상에서 운영되도록 구성되었기 때문에 보안에 대하여 프로그래머들에게 주관심사가 되고있다. 자바는 포인터를 사용하지 않고 또한 메모리에 대한 접근을 허용하고, Encryption기법을 사용하여 한층 강화된 보안체계를 가지고 있다. 또한, 바이트 코드를 번역할 때 자바언어에 위배되는 사항이 있을 때에는 운용이 되지 않도록 하여 바이러스로부터 보호를 받을 수 있게된다.

▶ 자바는 앞에서 기술한 것처럼 시스템에 종속적이지 않다.

왜냐하면, 자바 바이트 코드를 번역기를 통하여 번역하기 때문에 번역기가 있는 어떤 시스템이든지 운영이 가능하다. 그러므로, 하나의 플랫폼에서 개발한 프로그램을 네트워크를 통하여 다른 플랫폼에서 실행시킬 수 있다.

▶ 이식성이 뛰어나다.

왜냐하면 자바는 IEEE의 754 표준에 따라 만들어졌기 때문이다. 그리고, 기존의 언어가 실행 과일을 운용하기 위해 시스템의 환경을 검사하는데 시간을 많이 소비하는 것과는 다르게, 자바는 자바 바이트 코드를 번역기를 통하여 번역하여 프로그램을 운용하기 때문에 고성능이다. 또한 다중처리를 할 수 있도록 구성되어있다.

V. C++ 언어와 자바의 비교

자바는 C++에 기초를 두고 거의 사용하지 않는 부분과 많이 사용하는 부분의 상당 부분을 제거하여 더욱 객체 지향적인 모습을 가지고 있다. C++는 매우 강력한 객체 지향적 프로그래밍 언어이다. 하지만, 강력한 다른 언어와 마찬가지로 코드를 만듦에 있어서 프로그래머들을 어렵게 만든다. 특히, 코드의 유지 보수에 많은 시간과 노력을 빼앗기게 된다. 자바는 이러한 측면을 제거하여 다음과 같이 기존의 C 혹은 C++과 많은 차이를 다음과 같이 보이고 있다.

▶ 자바는 struct, union, pointer를 사용하지 않는다.

자바는 포인터를 가지지 않는다. 즉, 자바 언어는 참조에 의하여 모든 배열들과 객체들을 전달시킨다. 포인터를 사용하지 않기 때문에 포인터의 관리 소홀로 인한 문제점이 발생하는 것을 방지할 수 있다. 또한, 자바 언어는 struct와 union문을 지원하지 않는다. 그러면, C 언어와 자바 언어의 원시 파일을 통하여 비교를 해보겠다.

C 언어의 경우

```

.....
struct student {
    char name「NAMESIZE」;
    char course「COURSESIZE」;
    int point;
};

```

자바의 경우

```

.....
class Student{
    String name;
    String course;
    private int point;
};

```

▶ 자바 언어의 명령 라인 인수(Command Line Argument)가 C++와 다르다.

C 혹은 C++ 언어로 만든 프로그램을 처음에 작동시킬 때, 시스템은 두 가지 매개 변수 즉, argc 와 argv를 통과시킨다. 여기서, argc는 명령 라인 상의 인수의 수를 나타내고, argv는 그 인수를 포함하고 있는 문자 배열에 대한 포인터를 말한다. 하지만, 자바 언어에는 위의 경우와는 다르다. 자바 언어로 만든 프로그램을 시작할 때에는 단지 한 가지 매개 변수 즉, args만이 통과된다. 여기서의 args는 인수를 포함하는 문자 배열을 의미하는 것으로, 배열의 포인터를 가리키는 것은 아니다.

▶ 자바 언어의 문자열(String)은 첫 번째 클래스 객체이다.

자바는 java.lang 패키지의 부분으로 String class를 가진다. 이것은 기존의 C 혹은 C++에서 사용되는 문자열과 다르다. 기존의 언어인 C 와 C++에서의 문자열은 끝 부분이 항상 Null 문자(/0)에 의하여 채워지는 문자 배열이다. 하지만, 자바의 문자열은 항상 첫 번째 클래스 객체이다. 그리고, 마지막은 Null 문자로 채워지지 않는다. 그러면, C언어와 자바 언어의 소스를 통하여 비교를 해보겠다.

C 언어의 경우

```

.....
main(){
    char *s="Hare are you?";
    char t[]="Fine!";
    printf("%s, %s/n", s, t);
    GoCopy(s, t);
    printf("%s, %s/n", s, t);
}

```

```

GoCopy(char *string1, char *string2){
    for(; *string1 !='/0';string1++, string2++){
        *string2= *string1;
    }
}

```

자바의 경우

```

.....
class strepy {
    public static void main(String args[] ) {
        String s="How are you?";
        StringBuffer t=new StringBuffer("Fine!");
        System.out.println(s + " , " + t);
        GoCopy(s, t);
        System.out.println(s + " , " + t);
    }
    static void GoCopy(String string1, StringBuffer string2) {
        int i, length=string1.length();
        for( i=0;i < length;i++)
            string2.setCharAt(i, string1.charAt(i));
    }
}

```

▶ 자바 언어는 메모리 관리를 스스로 한다.

기존의 언어인 C 혹은 C++ 언어에서는 메모리에 대한 관리를 프로그램 내에서 정하여야 한다. 즉, malloc 와 free등을 지정하여야만 된다. 이것은 프로그램에 대한 해킹을 가능하게 하고, 많은 버그(Bug)를 만들게 하는 요인이기도 하다. 하지만, 자바에서는 memory에 대한 할당 및 해제를 자동으로 수행한다. 즉, 프로그램에 메모리에 대한 관리를 지정할 필요가 없다. 이것을 통하여 프로그램은 좀 더 보안적이며 안정성을 가질 수 있게 되었

다. 이러한 메모리의 자동 관리를 가능하게 하는 것은 Garbage Collection 기법 때문이다. 자바에서는 객체가 만들어질 때 자동으로 메모리를 할당하고, 객체가 더 이상 사용되지 않을 때 Run-Time Garbage Collector에 의하여 메모리가 해제된다.

▶ 자바의 데이터 유형은 항상 고정되어 있다.

기존의 언어인 C 혹은 C++에서는 시스템 혹은 오퍼레이팅 시스템에 따라 데이터의 유형이 변한다. 또한, unsigned 데이터 유형이 존재한다. 하지만, 자바에서의 데이터 유형은 하드웨어 혹은 오퍼레이팅 시스템과는 무관하게 항상 고정되어 있다. 자바 언어의 데이터 유형은 단순형과 복합형이 있다. 단순형에는 integer, floating, point, boolean, character, byte등이 있고, 복합형에는 배열, 클래스, 인터페이스의 세 가지가 있다. 자바 언어의 데이터 유형에 대하여는 뒤에서 자세히 언급하도록 하겠다. 또한, 자바 언어에서 라이브러리로 제공하고 있는 java.lang 클래스에 이러한 데이터 유형에 대한 많은 Method들이 있다.

▶ 자바 언어에서의 Boolean 값은 항상 true 혹은 false이다.

자바 언어는 기존의 언어처럼 Boolean 값에 정수를 돌려주지는 않는다. 자바 언어의 이러한 논리형은 숫자 값을 가질 수도 없고, 숫자 값으로 형 변환을 할 수도 없다. 하지만, 기존의 다른 언어에서는 숫자 값을 가질 수도 있고, 숫자로의 형 변환도 가능하다. 이것은 프로그램을 취약하게 만들 수 있는 것으로 자바 언어에서는 이런 문제점을 제거하였다.

▶ 자바는 typedef와

define을 지원하지 않는다.

왜냐하면, 자바 언어에는 프리-프로세서(Pre-Processor)가 없기 때문이다. 그래서 define이나 macro와 같은 것을 사용할 수 없다. 또한, 자바 언어는 기존의 언어와는 다르게 번역하는 언어이기 때문이다.

▶ 자바는 단일 상속만을 지원한다.

자바 언어에서는 단일의 상속(Inheritance)만이 있고, 다중의 상속은 지원되지 않는다. 만약 기존의 언어에서처럼 다중의 상속을 반드시 사용해야 하는 경우에는 인터페이스를 이용하면 된다.

VI. 자바 패키지

객체 지향적 프로그래밍의 가장 큰 장점은 수정 혹은 보완을 필요로 할 때 기존에 만들어 놓은 객체들을 재사용 할 수 있다는 점이다. 객체 지향적 언어인 자바 언어에서는 개발자들을 위하여 관련성이 깊은 유사한 객체들을 라이브러리들로 만들어 놓아 개발하기 쉽고, 수정 혹은 보완이 용이하도록 만들어 놓았다. 이것을 자바 언어에서는 자바 클래스 라이브러리라고 하고, 이런 클래스 라이브러리들을 패키지(Package)라고 부른다.

자바 언어의 패키지는 클래스와 인터페이스의 묶음이고, 많은 이름들을 관리하고, 서로의 충돌을 피하는도록 하는 도구이다. 모든 클래스와 인터페이스의 이름은 같은 패키지에 포함되어 있다. 패키지의 이름은 마침표로 구분한다(period-separated words). 첫 번째 부분의 이름은 패키지를 개발한 구조를 나타내도록 되어 있다.

자바에서 제공하는 이러한 패키지들은 자바 언어의 핵심이라고 할 수 있는 클래스들을 제공한다. 개발을 하게 될 때, 개발자들은 이러한 패키지에서 제공하는 클래스들과 인터페이스들을 사용할 수 있고, 이 패키지를 사용하여 프로그램을 만드는 것이 새로이 클래스와 인터페이스를 만들고 시작하는 것 보다 훨씬 쉽고 용이하다.

▶ 자바 언어 패키지(java.lang)

자바 언어 패키지는 자바 언어의 핵심을 이루는 클래스들을 제공한다. 이러한 패키지는 다시 여러 개의 그룹으로 묶여져 있다. 그것은 객체, 데이터 유형 분류, 문자열, 시스템과 자원, 쓰레드, 클래스들, 산술, 예외 와 오류 처리 및 프로세서의 그룹

으로 구성된다.

▶ 자바 입출력 패키지 (java.io)

자바 입출력 패키지는 데이터 혹은 파일들을 읽고 쓰는데 사용되는 입력 스트림과 출력 스트림의 모음으로 제공이 되고 있고, 이것은 java.io 클래스 안에서 다시 입력 스트림, 출력 스트림 과 오류 등과 같은 나머지의 세 가지 구분으로 나누어진다.

▶ 자바 유틸리티 패키지 (java.util)

자바 유틸리티 패키지에는 데이터 구조 클래스, 날짜, StringTokenizer, 속성, 관찰자(Observer), Random-Number 생성기, 열거(Enumeration)등으로 구성된다. 이 패키지의 구조는 상위에 java.lang.Object가 있다.

▶ 자바 네트워크 패키지 (java.net)

자바 패키지 안에는 인터넷 상에서 운영되는 TCP/IP, URL, Ftp 등과 같은 프로토콜을 지원하기 위해 네트워크에 관련된 패키지를 포함하고 있다. 이러한 자바 네트워크 패키지는 TCP/IP 소켓 혹은 UDP 소켓으로 운영되는 클라이언트/서버 환경을 구현할 수 있는 손쉬운 방법을 제공할 뿐만 아니라, 인트라넷(IntraNet)상에서의 통신과 인터넷 상에서의 통신을 지원하고 있기 때문에, 손쉽게 전 세계를 연결하는 기업의 네트워크 환경을 구축할 수 있도록 한다. 특히, 인터넷을 위하여 현재 인터넷에서 사용되고 있는 프로토콜들 외에 SNMP, TFTP, VMTP, ICMP, ATM등을 지원하고 있다.

▶ 자바 애플릿 패키지 (java.applet)

오디오와 관련된 AudioClip과 같은 자바 애플릿들을 가진 패키지이다. 이 패키지의 구조는 상위에 java.lang과 java.awt가 있다.

▶ 자바 추상 윈도우 툴킷 패키지 (java.awt)

자바 추상 윈도우 패키지에는 그래픽 환경에서 사용하는 이미지, 메뉴, 폰트, 체크 박스, 버튼 등과 같은 패키지들을 포함하고 있어서, 손쉽게 사용자의 환경을 그래픽화 시킬 수 있는 기능을 제공하

고 있다.

VII. 자바의 성장

현재까지의 인터넷의 역사를 살펴보면, 텍스트 중심의 데이터 이용에서 웹이 등장함으로써 GUI 환경으로 이동을 하였고, 또한 인터넷에서 사용하는 인터페이스 및 프로토콜에 대한 표준을 제공하였다. 특히 그래픽을 이용한 화면의 제공과 데이터의 이용 방법은 사용자들에게 새로운 인터넷의 세계를 제공하여 급속한 발전을 가져왔다.

하지만, 이러한 웹의 진정한 이용은 사용자의 환경 즉, 사용하는 컴퓨터가 어느 것이든지 혹은 오퍼레이팅 시스템이 어느 것이든지 상관없이 하나의 공통된 프로그램을 이용 가능해야 하는 것이다. 하지만, 지금까지의 웹은 프로그램을 운영할 수 있는 환경을 제공하지 못하고 데이터의 양이 증가함으로써 기존의 사용자 네트워크에 병목현상을 가져 왔다. 또한, 사용자의 계속되는 요구의 증가를 기존의 언어로는 지원할 수 있는 한계에 이르렀다.

이러한 웹이 가지고 있는 문제를 해결한 것이 자바 언어이다. 자바 언어가 가진 프로그래밍 상의 특성과 대화형, 보안성 등의 특징은 현재 사용자들의 욕구를 충족시킬 수 있는 기능의 제공만이 아니라 인터넷의 웹을 통하여 데이터의 전송뿐만 아니라 상호 작동 적인 실시간 응용 프로그램을 운영할 수 있다는 것이다. 한 걸음 더 나아가 자바 언어는 사용자의 시스템 혹은 오퍼레이팅 시스템이 어느 것이든지 상관없이 하나의 소스를 가지고 단지 자바 언어를 이해하는 브라우저만 있게 되면, 인터넷을 통하여 언제든지 어디서든지 이용이 가능하다는 것이다.

그리고, 자바 언어는 기존의 C 혹은 C++언어와 비슷하여 배우기가 쉬운 특징도 가지고 있다. 단지 데이터의 유형 검사, 자동 스토리지 관리, 가배지 컬렉션과 같은 기법의 이용 등이 다를 뿐이다. 이런 모습은 기존의 Lisp언어와 유사한 것이기도 하다. 또한, Lisp이 번역 가능한 언어이듯, 자바 언

어도 번역 가능한 언어이다. 3차원을 지원하기 위한 VRML이 그래픽 렌더링을 위한 특수 언어이지만, 자바 언어는 범용성을 갖는 언어이다. 자바 언어는 앞으로 C 혹은 C++ 언어를 대체할 것이다.

▶ 네트워크 컴퓨터와 자바 기반의 테스웍 컴퓨터 OS인 코나(Kona)

현재 업계에서 주목받고 있는 네트워크 컴퓨터인 인터넷 터미널을 작성하는 언어로 사용할 계획이다. 자바 언어는 초기에는 기업 내부의 네트워크와 인터넷을 연계한 인트라넷(IntraNet)에서 운영 가능한 응용 프로그램을 중심으로 발전하여, 추후에는 네트워크 컴퓨터들의 마이크로 커널을 만드는 언어로 사용될 것이다. 현재, 네트워크 컴퓨터의 운영체계가 이미 자바 언어로 만들어져 있고, 그것은 코나(Kona)라고 불린다. 현재 썬, 오라클, IBM이 각각 생산해낼 네트워크 컴퓨터의 표준 명세를 공동으로 설계하기로 하였고, 애플, IBM, 노벨, HP 등 9개의 OS 개발 업체들이 자사의 플랫폼에 코나의 운영이 가능하도록 지원할 것임을 밝히고 있어, 아마도 썬에서 자바 언어로 만든 코나 OS는 네트워크 컴퓨터의 표준으로 될 것이다.

이제, 기업에서는 공통된 응용 프로그램을 사용하기 위하여 더 이상 상이한 시스템들과 많은 오퍼레이팅 시스템의 버전들을 위한 프로그램들을 가질 필요가 없이, 자바로 만들어진 하나의 소스를 컴파일 하여 실행 가능한 응용 프로그램을 사용할 수 있는 기반을 가지게 된 것이다.

현재 자바 언어는 대화형 텔레비전 세트, VCR, 토스터 등의 가전 제품 등의 프로그래밍 언어로 사용이 되고 있다. 가장 최근에는 자바 칩(Chip)이 발표되어 컴퓨터 및 각종의 전자 제품에서 사용될 예정이다. 멀지 않은 장래에 여러분은 자바 칩을 탑재하고 기본적인 인터넷 연결 프로그램을 갖춘 네트워크 컴퓨터인 자바 스테이션과 같은 것을 만나게 될 것이고, 자바 칩을 이용한 핸드폰, PDA, 게임기 등을 이용하게 될 것이다. 그리고, 핸드폰의 액정을 확장하는 경우에는 웹 서핑이 가능할 수도 있다. 이것은 인터넷 라인만 연결이 되면 바로 인터넷 상에서 모든 것을 사용할 수 있도록 자바 칩

이 디자인되었기 때문이다. 현재 자바 칩은 피코자바(PICOJAVA), 마이크로자바 및 울트라자바칩이 나와있다.

지금 현재 컴퓨터 업계에서는 네트워크 컴퓨터에 대한 관심이 고조되어 가고 있다. 이는 네트워크 컴퓨팅과 함께 새로운 기술로 인식되어 오고 있다. 네트워크 컴퓨터는 모든 프로그램을 자신의 하드 드라이브에 저장하던 현재의 PC와는 다르게, 필요한 최소한의 프로그램들을 가지고 있고, 나머지는 인터넷을 통하여 서버로부터 다운 로드하여 사용하는 컴퓨터이다. 그러나, 이것에 의문점이 생기게 된다. 그것은 인터넷을 이용하여 프로그램을 다운 로드하여 사용해야 한다면, 사용을 위해 대기하는 시간이 길어지지 않을까 하는 것이다.

하지만, 자바는 프로그래밍의 모델을 새로이 제공하고 있다. 즉, 자바 언어는 고도로 압축된 코드를 만들어 냄으로서 인터넷 상에서 빠르게 인터넷 상에서 프로그램을 다운 로드하여 사용할 수 있도록 한다. 또한, 번역할 수 있는 코드는 실행 파일보다도 크기가 작다. 그리고, 자바 언어는 객체 지향적 언어이기 때문에 모듈화가 쉬워진다. 즉, 인터넷을 통하여 필요한 부분을 그때마다 다운 로드를 받을 수 있기 때문에, 전체의 프로그램을 다운 로드 하지 않아도 된다는 것이다. 이러한 이유로 자바 언어를 이용하는 네트워크 컴퓨터가 기업 내에서 사용하는 많은 개인용 컴퓨터를 대체하게 될 것이다.

자바 언어는 전자 제품, 인터넷 및 인트라넷을 위한 가장 적합한 해결책이 될 것이고, 점점 사라지고 있는 언어로 많은 기능들이 신속하게 포함되고 있다. 그러므로, 현재는 약간의 미해결된 부분들도 있지만 조만간 완전한 언어 및 많은 보조 프로그램들을 가지는 프로그래밍 언어가 될 것이다. 자바 언어는 네트워크 컴퓨팅을 위한 가장 최적의 해결책이다.

VIII. 자바 관련 웹 사이트

현대에는 정보의 사회이고, 어느 누가 필요한 시기

에 적절한 정보를 가지고 있는가하는 것이 승패의 열쇠이다. 자바 언어는 새로이 등장한 객체 지향적 프로그래밍 언어이다. 또한, 자바 언어는 성장하는 언어이다. 이러한 언어를 배우기 위하여 우리는 항상 정확하고 신속하게 정보를 확보해야만 한다. 이를 위해서, 자바 언어와 관련된 웹 사이트를 소개하는 것으로 자바 언어에 대한 소개를 마치도록 하겠다. 자바 언어는 전 세계의 컴퓨터 인구가 동시에 제공받은 프로그래밍 언어이다. 그러므로, 국내의 프로그램들이 세계적인 프로그램으로 성장할 수 있는 좋은 기회이기도 하다. 그러므로, 국내의 프로그래머들은 정확한 정보를 가지고 자바 언어

에 대한 분석 및 좋은 프로그램 만들기에 주력하였으면 한다. 다음의 웹 사이트로부터 자바 컴파일러, 소스, 명세서, 관련 백서 및 다양한 응용 프로그램, 자바 관련 상업용 프로그램 등에 대한 정보를 얻을 수 있을 것이다.

- * 자바 홈페이지 : <http://www.Javasoft.com/>
- * 썬 홈페이지 : <http://www.sun.com/>
- * 자바월드 메가진 홈페이지 : <http://www.sun.com/javaworld>
- * 자바코리아 사용자그룹 홈페이지 : <http://voyager.kti.co.kr>

저 자 소 개



金 泰 賢

1963年 7月 25日生

1988年 8月 서강대 졸

1990年 7月~1994年 5月	현대전자
1994年 5月~1994年 12月	한국 오라클
1995年 1月~현재	한국 썬.마이크로시스템즈 SE

주관심 분야 : Network & Internet