

최신 모바일 멀티미디어 콘텐츠의 제작환경: Java ME

김일민* · 조세홍* · 박재희**

1. 서론

자바 언어는 1995년 Sun사에 의해 처음 소개되었으며, 객체 지향 프로그래밍 언어 및 인터넷 분산 환경의 개발 언어로써 많이 사용하게 되었다. 자바 프로그램은 바이트 코드로 번역되어 실행됨으로써, 특정 플랫폼을 대상으로 개발된 응용 프로그램을 다른 플랫폼으로 포팅할 때 발생하는 문제점을 해결하였다. 특히 인터넷, 네트워킹, 데이터 베이스 접속, 멀티 스레딩에 관한 다양한 메소드를 제공함으로써 자바 언어를 사용하는 개발자의 수는 급격히 증가하였다. 그러나 자바를 사용하는 개발자의 수가 급증함에 따라서, 다양한 개발자의 요구를 하나의 자바 개발 킷만을 가지고 충족할 수 없음을 알게 되었다.

이러한 요구에 따라 Sun사에서는 자바 표준 에디션 Java SE(Standard Edition), 서버의 기능을 강화한 업무용 개발 툴인 Java EE(Enterprise Edition), 자바를 경량화시켜 모바일 환경에서 응용 프로그램 개발이 가능하도록 만든 Java ME (Micro Edition)을 개발하였다. J2ME는 1999년 Java One Conference에서 처음 발표되었으며, 자바 2를 가진 제품 및 휴대용 컴퓨팅 장치에서 실행 가능하도록 만든 것이다. 다음 그림 1은 Sun사의

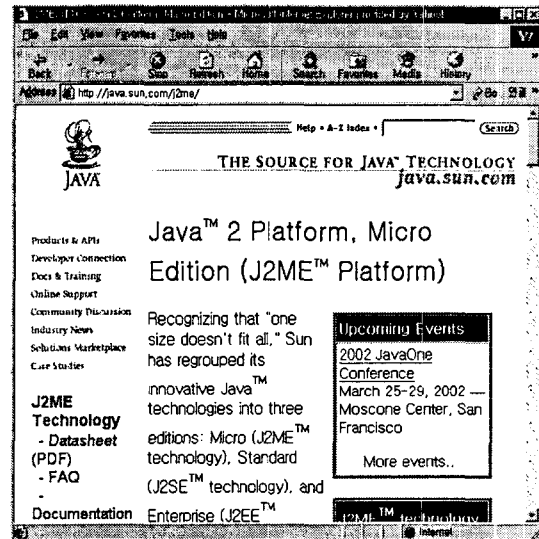


그림 1. Java ME 홈페이지

Java ME 페이지(<http://java.sun.com/j2me/>)를 캡처한 것이다.

1.1 WAP/WML/i-Mode

현재 핸드폰이나 PDA를 사용, 인터넷에 접속하여 날씨, 주가와 같은 정보를 얻을 수 있다. WAP(Wireless Application Protocol)은 1995년 처음 발표되었으며, 이동전화와 같은 무선 단말기 상에서 인터넷을 이용할 수 있게 해주는 개방 표준 프로토콜이다. WAP은 이동전화 단말기에 설치된 서버를 경유해 인터넷 패킷을 무선 통신망에

*한성대학교 컴퓨터 공학부
 **계명대학교 컴퓨터 전자 공학부

전송하는 방식이다. WML(Wireless Markup Language)은 HTML과 유사한 마크업 언어이며, 콘텐츠를 특정지으며 핸드폰과 같은 제한된 성능을 가진 장치의 인터페이스를 위해 개발되었다. I-mode는 WAP를 대치하기 위해서 NTT DoCoMo 사에 의해 처음 발표되었다. i-Mode는 cHTML(Compact HTML)이라는 마크업 언어를 포함하고 있으며, cHTML은 기본적인 HTML의 일부에 NTT 고유의 태그나 심볼을 추가한 것이다.

WAP와 i-Mode는 웹 서핑을 위한 응용에는 아무런 문제가 없지만, 취약한 보안성, 다소 불편한 사용자 인터페이스, 서버와 연결상태를 유지해야 한다는 문제점이 있다. 특히 무선 네트워크의 대역폭이 증가하고, 휴대폰의 성능이 향상됨에 따라, 보안성에 문제가 없으며, 기능이 더욱 복잡해진 클라이언트/서버와 같이 응용을 요구하게 되었다. J2ME는 이러한 요구에 부응하고, WAP 기반 응용 프로그램의 한계를 극복하기 위해서 설계되었다.

1.2 휴대용 컴퓨팅 장치

J2ME는 휴대폰이나 PDA와 같은 내장형 기기에서 동작하는 자바 프로그램을 개발하기 위한 자바 개발 킷이다. 이러한 기기들은 일반 PC보다 적은 용량의 주기억장치와 CPU 컴퓨팅 파워가 낮으며, 디스플레이 장치의 용량도 매우 제한적이라는 특징이 있다. 이러한 환경에서 당연히 적은 크기의 JVM(Java Virtual Machine)에서 제한적인 기능을 가진 응용 프로그램이 실행된다.

일반 개인용 컴퓨터의 주기억장치 용량은 128MB~256MB 정도이지만, PDA의 경우 주기억장치의 용량이 8MB~16MB 정도이다. 그러므로 휴대용 컴퓨터에서 실행되는 프로그램의 크기는 8MB로 제한된다. 또한 보조 기억장치의 경우

일반 개인용 컴퓨터는 20GB~80GB 정도이지만, 휴대용 컴퓨터 장치의 보조 기억장치는 500MB 정도이므로 그 용량의 차이가 매우 크다.

일반 컴퓨터의 입력 장치는 키보드, 마우스 등 다양한 입력 장치를 사용할 수 있지만, 휴대용 컴퓨팅 장치의 경우 입력 장치 또한 제한적이다. PDA의 경우는 펜을 사용한 터치 스크린이 일반적이다. 터치 스크린은 입/출력을 위해 동시에 사용될 수 있으므로 공간을 절약할 수 있다. 텍스트를 입력하기 위해서 키패드를 사용하는 것도 가능하다. 키패드는 원래 숫자를 입력하기 위한 것이나, 원하는 알파벳이 나올 때까지 동일한 키를 여러 번 누름으로써, 알파벳을 입력하는 것도 가능하다. 이러한 방법은 대용량의 텍스트를 입력하는데에는 적절하지 않지만, 주소와 같이 간단한 데이터를 입력하는 것은 가능하다. 보다 세련된 입력 방식은 음성 인식 기법을 이용할 수도 있지만, 아직은 보편적으로 채택되고 있지는 않다.

휴대용 컴퓨팅 장치에서 가장 많이 사용되는 출력장치는 액정 화면이다. 일반 컴퓨터에서는 대형 액정 화면을 사용하는 것이 가능하지만, 휴대용 컴퓨팅 장치의 액정 화면의 크기는 매우 작다. 소형 화면의 경우 4~5줄 정도의 텍스트를 출력할 수 있으며, 대형 화면의 경우 20줄 정도를 출력할 수 있다. 또한 문자의 크기와 제공되는 폰트의 수가 매우 제한적일 수밖에 없다.

2. Java Micro Edition

Sun사는 기존의 자바 실행 환경이 대용량의 자원을 필요로 하므로 휴대용 컴퓨팅 장치와 같은 제한적인 하드웨어에 적합한 새로운 자바 가상 머신을 구현하였다. J2ME는 무선 응용 프로그램 개발을 위해 다음과 장점을 가진다. 자바언어는 플랫폼에 독립적이며, 풍부한 네트워킹 기능 및

GUI(Graphic User Interface)를 내장하고 있다. 또한 자바는 보안 모델을 가지고 있으며, 동적으로 프로그램을 배포할 수 있다. J2ME는 무선 네트워크를 통해서 손쉽게 응용 프로그램을 다운로드하고 설치할 수 있는 동적 배포 메커니즘을 제공한다.

2.1 KVM의 특징

J2ME의 새로운 가상 머신 이름은 KVM으로 명명되었으며, 소형 컴퓨팅 장치에서 실행을 최적화하였다. 그러므로 J2ME는 기존의 32비트 주소 공간의 JVM이외에 제한적인 메모리를 제공하는 16비트(또는 32비트) 아키텍처의 KVM을 지원한다. 여기서 K는 Kilobyte를 나타내며, KVM의 설계 목표가 킬로바이트 단위의 소형 가상 머신임을 의미한다. KVM의 특징을 간단히 정리하면 다음과 같다.

1) 바이트 코드의 검증이 생략되었다. 바이트 코드에 대한 검증작업은 많은 자원과 시간이 소요되는 작업이므로, 바이트 코드에 대한 검증은 소스 코드를 컴파일하는 데스크 탑 컴퓨터에서 실행한다.

2) long형 정수와 부동형 실수는 기본 데이터 유형에서 생략되었다. 물론 특수 목적의 응용 프로그램에서는 필요하므로 필요에 따라 추가하는 것은 가능하다. 일반적으로 휴대용 컴퓨팅 기기는 하드웨어적으로 부동 소수점을 지원하지 않으며, 소프트웨어적으로 지원하기에는 오버헤드가 너무 크기 때문이다.

3) 사용자가 객체의 finalize() 메소드를 정의하지 못하게 하였다. J2SE의 경우 finalize() 메소드를 호출함으로써, 이 객체가 가비지 컬렉터에 의해 수거되어야 할 객체임을 표시할 수 있다. J2ME의 가비지 컬렉터는 기능이나 성능이 부족하지만,

보다 적은 메모리를 사용한다.

4) JNI(Java Native Interface)를 지원하지 않는다. 자바 프로그램에서 자바언어로 작성되지 않은 라이브러리 사용해야 할 경우에 JNI를 이용한다. J2SE에서는 입출력이나, 그래픽 작업 등과 같은 시스템 호출을 필요로 하는 네이티브 메소드를 호출할 수 있지만, J2ME에서는 더 이상 지원되지 않는다.

5) 예외처리가 제한적이다. 모든 예외 클래스를 지원하는 것은 저장공간이 낭비를 초래하며, 대부분의 내장형 기기들은 오류를 무시하기 때문에 그렇게 할 필요도 없다. 그러므로 KVM에서는 다음의 세 가지 오류만을 처리한다.

- java.lang.Error
- java.lang.VirtualMachineError
- java.lang.OutOfMemoryError

6) 2차원 이상의 배열은 기본적으로 사용할 수 없다. 대부분의 응용 프로그램은 1차원 배열만을 사용하므로, 컴퓨팅 오버헤드가 큰 2차원 이상의 배열은 지원하지 않는다.

7) J2SE와 비교하여 제한적인 보안모델을 제공한다. 자바 바이트 코드가 자비 힙 메모리 이외의 부분을 참조할 수 없으며, 응용 프로그래머는 사전에 정의된 자바 API만을 사용하여야 한다.

J2ME는 소형 장치에서 실행되므로, 기존의 실행환경의 클래스에서 불필요하다고 판단되는 클래스들을 제거하였다. 일부 J2SE 클래스의 메소드를 제거하거나 기능을 단순화하여 KVM의 용량을 경량화하였다. 또한 J2ME 설계자들은 소형 장치에 적합한 새로운 클래스를 추가하였으며, 기존의 클래스를 재정의하여 J2SE에서는 없었던 기능들을 추가하였다. 또한 J2ME는 자바 실행 환경을 휴대 컴퓨팅 하드웨어에 따라 커스터마이징하기 위해서 configuration과 profile을 사용한다.

2.2 Configuration

Configuration은 유사한 컴퓨팅 능력을 가진 장치에서 실행될 J2ME 기본 클래스와 자바 가상 머신을 정의한다. 모든 휴대용 장치가 동일한 컴퓨팅 능력을 가지고 있지 않으므로, J2SE와는 달리 휴대용 장치의 용량에 따라 포함되는 기본 클래스를 달리 하는 것이 필요하다. 지금까지의 configuration은 CLDC(Connected Limited Device Configuration)와 CDC(Connected Device Configuration) 두 가지가 정의되어 있다. CLDC는 휴대폰과 같은 보다 소규모 장치의 자바 가상 머신을 정의하고 있으며, CDC는 PDA와 같은 중대형 장치의 자바 가상 머신을 정의하고 있다. 각 configuration은 지원해야하는 최소한의 클래스와 API를 정의하고 있다.

자바 응용 프로그램	
KVM + CLDC API	Native API
Native OS	
하드웨어	

그림 2. CLDC의 구성

CLDC에서 지원되는 API는 기존 J2SE에서 java.lang, java.util, java.io 패키지의 일부 클래스이다. 이 패키지의 API들도 많은 부분이 생략되거나, 재정의되었으므로, 기존의 J2SE 자바 프로그램은 거의 실행되지 않는다. 다음은 CLDC 기기에서 공통적으로 사용가능한 클래스를 나열한 것이다.

- 1) java.lang 패키지
- java.lang.Object
- java.lang.Class
- java.lang.Runtime

- java.lang.System
- java.lang.Thread
- java.lang.Runnable
- java.lang.String
- java.lang.StringBuffer
- java.lang.Throwable
- java.lang.Boolean
- java.lang.Byte
- java.lang.Short
- java.lang.Integer
- java.lang.Long
- java.lang.Character
- java.lang.Math

2) java.util 패키지

- java.util.Vector
- java.util.Stack
- java.util.Hashtable
- java.util.Enumeration
- java.util.Calendar
- java.util.Date
- java.util.TimeZone
- java.util.Random

3) java.io 패키지

- java.io.InputStream
- java.io.OutputStream
- java.io.ByteArrayInputStream
- java.io.ByteArrayOutputStream
- java.io.DataInput
- java.io.DataOutput
- java.io.DataInputStream
- java.io.DataOutputStream
- java.io.Reader
- java.io.Writer
- java.io.InputStreamReader

```
java.io.OutputStreamWriter
java.io.PrintStream
```

CDC는 현재 512KB의 ROM과 256KB의 RAM 및 네트워킹 접속 장치를 포함하는 것을 최소 사양으로 하고 있다. CDC는 고성능의 PDA, 자동차 항법 시스템(car navigation system)과 같은 장치에서 실행되며, 완벽한 자바 가상 머신이 지원된다. CDC API는 CLDC 명세에서 정의된 모든 API를 포함하며, 파일 주소를 이용한 프로토콜과 JAR(Java Archive)를 이용한 프로토콜을 지원한다. CDC에 대한 최신 정보는 <http://java.sun.com/products/cdc/> 페이지를 참조하기 바란다. CLDC는 160KB에서 512KB의 메모리와 16 비트 이상의 프로세서를 가진 휴대 전화, 호출기, 소형 PDA에서 실행된다. 또한 제한된 대역폭으로 네트워크에 연결할 수 있어야 한다. CLDC에 대한 최신 정보는 <http://java.sun.com/products/cldc/> 페이지를 참조하기 바란다.

자바 응용 프로그램	
Classic VM + CDC API	Native API
Native OS	
하드웨어	

그림 3. CDC의 구성

2.3 Profile

Profile은 특정 J2ME configuration에 원하는 도메인의 클래스를 추가한 것이다. 그러므로 profile은 configuration과는 달리 응용 분야에 따라 포함되는 클래스가 달라진다. 즉 profile은 특정 응용 분야나 특정 장치의 활용의 필요성에 따라 포함되는 클래스를 결정한다. Profile의 장점은 원하는 클래스만을 추가함으로써 최소한의 용량에

필요한 기능을 수행할 수 있다. 그러나 응용 프로그램이 특정 profile을 가진 하드웨어에서만 작동하므로 프로그램의 이식성이 떨어진다. 그림 4는 Profile의 구성을 나타내었다.

자바 응용 프로그램
Profile
Configuration
Native OS
하드웨어

그림 4. Profile의 구성

J2ME의 표준안은 JCP(Java Community Process)과의 협의로 이루어지고 있다. JCP는 자바 기술이 Sun사에 의해 독점되지 않고, 커뮤니티의 합의하에 따라 개방화된 표준기술로 발전할 것을 보장하기 위해 만들어졌다. 현재 JCP에서 여러 가지 profile 표준을 작성 중에 있으며, 그 중 몇 가지를 소개하면 다음과 같으며, 이를 정리하면 표 1과 같다.

① Foundation profile

- CDC 기반하에서 J2ME를 지원하는 가장 기본적인 명세이다. 다른 프로파일들은 foundation profile을 기반으로 작성될 수 있다.

② Personal profile

- 기존의 Personal java로 작성된 응용 프로그램과의 호환성을 위한 profile이다.

표 1. J2ME의 Profile

Profile	기반 configuration	ROM	RAM
Foundation	CDC	1024K	512K
Personal	CDC+Foundation	2.5M	1M
PDA	CLDC	256KB	256KB
MIDP	CLCD	128KB	32KB
RMI	CLDC+Foundation	2.5M	1MB

③ PDA profile

- 소형 PDA를 위해 개발된 profile이다. J2SE AWT의 일부 GUI를 지원하며, 512KB - 16MB의 메모리를 가지는 PDA를 위해 개발되었다.

④ MIDP(Mobile Information Device Profile)

- CLDC를 기반으로 하고 있으며, 휴대전화, 호출기, PDA 등을 위한 profile이다. 128KB의 ROM과 32KB의 RAM을 최소 사양이다. 일반적인 자바 모바일 프로그래밍은 MIDP를 기반으로 작성된다.

⑤ RMI profile

- 휴대용 컴퓨팅 장치에서 RMI와 JINI를 지원하기 위해 개발된 profile이다. RMI는 자바의 분산 객체 기술로써 원격 메소드 호출을 의미하며, JINI는 플러그인이 가능한 동적 네트워크 장치 기술이다.

앞에서 설명한 많은 프로파일 중에서 MIDP가 가장 먼저 발표된 프로파일이며, 많은 모듈라, LG, Samsung 휴대전화 제품들이 MIDP를 채택하여 출시되고 있다. 현재 J2ME의 MIDP가 지원되는 이동 컴퓨팅 하드웨어를 정리하면 다음과 같다. Motorola는 2001년 4월 J2ME가 가능한 iDEN 3000 휴대폰을 생산하였다. LG 텔레콤은 2000년 10월부터 J2ME를 지원하는 i-Book 무선

단말기를 지원하기 시작하였다. 일본 최대의 무선망 사업자인 NTT DoCoMo는 2001년 2월 J2ME가 가능한 i-mode 서비스 휴대폰을 제공하기 시작하였다. 그러나 LG텔레콤과 NTT DoCoMo에서 구현된 MIDP는 MIDP 1.0의 사양이 발표되기 전에 구현되었으므로 J2ME의 표준을 정확히 준수하지는 않는다.

3. J2ME Wireless Toolkit

MIDP 장치를 위한 자바 응용프로그램을 개발하기 위해서 Sun사에서 제공되는 것이 J2ME Wireless Toolkit(J2MEWTK)이다. J2MEWTK은 MIDP 응용 프로그램 개발을 지원하며, 바이트코드 검증기, MIDP장치 에뮬레이터, 기본 IDE, Forte for Java와 통합 컴포넌트로 구성된다. J2MEWTK를 이용하면, PC에서 MIDP 응용 프로그램을 편집, 컴파일, 패키징(Jar 사용), 실행, 디버깅하는 것이 가능하다. MID(Mobile Information Device)에서 실행되는 자바 응용 프로그램을 "MIDlet"이라고 한다.

윈도우용 J2MEWTK 1.03 버전을 설치하기 위해서는 윈도우 98(Second Edition), 윈도우 NT 또는 윈도우 2000이 설치되어 있어야 하며, JDK 1.3

표 2. MIDP를 지원하는 하드웨어

제조회사	종류	profile	특징
Motorola i85s	iDEN	J2ME/MIDP	mobile phone, radio, micro-browser 내장
Motorola i50sx	iDEN	J2ME/MIDP	mobile phone, radio
Nokia 9210	GSM900	PersonalJava/MIDP	WAP web browser, fax
LG p510	CDMA	CLDC/kittyhawk	최대 10 jar파일, 최대 jar 파일 크기: 60KB
LG p520	CDMA	MIDP	
Samsung X-130	CDMA	MIDP호환XVM	Cleanroom 방식 구현
DoCoMo P503i	i-mode(PDC)	CLDC/DoComo Lib	SSL, 최대 7 jar파일, 최대 jar 파일: 10KB
DoCoMo F503i	i-mode(PDC)	CLDC/DoComo Lib	SSL, 최대 50 jar파일, 최대 jar 파일: 10KB
DoCoMo N503i	i-mode(PDC)	CLDC/DoComo Lib	컬러 액정

이 설치되어 있어야 한다. 또한 166MHz 이상의 CPU 속도, 64MB이상의 주기억장치와 15MB 이상의 디스크 공간을 필요로 한다. J2ME Wireless Toolkit는 <http://java.sun.com/products/j2metoolkit/download.html> 페이지에서 다운 받을 수 있다.

3.1 MIDP 응용 프로그램의 구조

MIDP는 CLDC에서 정의된 API와 MIDP에서 정의된 API를 사용할 수 있다. CLDC는 J2SE의 java.lang, java.io, java.util 패키지외에 javax.microedition.io 패키지를 포함한다. 이는 MID에서 사용하는 일반적인 네트워크 API를 제공한다. MIDP에서 사용 가능한 API를 그림으로 표현하면 그림 5와 같다.

1) 응용 프로그램 관리

javax.microedition.midlet 패키지에는 내장 응용 프로그램 관리를 위한 클래스들이 정의되어 있다. 모든 MIDP 응용 프로그램들은 이 패키지의 MIDlet 클래스를 상속하여야 하며, startApp(), pauseApp(), destroyApp() 메소드를 재정의하여

야 한다. 또한 인자가 없으며, public 접근자를 가진 생성자를 정의하여야 한다. 다음은 javax.microedition.midlet.MIDlet 클래스의 원형을 나타낸 것이다.

```
public abstract class MIDlet extends Objects {
    protected MIDlet();
    protected abstract void startApp()
        throws MIDletStateChangeException;
    protected abstract void pauseApp();
    protected abstract void destroyApp(boolean uncond)
        throws MIDletStateChangeException;
    public final String getAppProperty(String key);
    public final void notifyDestroyed();
    public final void notifyPaused();
    public final String getAppProperty(String key);
    public final void resumeRequest();
}
```

2) 네트워크 클래스

CLDC에서 정의된 Generic Connection framework에서 커넥션 인터페이스가 정의되어 있으며, MIDP에서 구현되어 있다. 커넥션 인터페이스를 구현한 클래스들은 javax.microedition.io 패키지에 포함되어 있다. 특히 MIDP상에서 Http Connection 인터페이스가 구현되어 있으므로 모든 MIDP용 장치에서 HTTP 통신이 가능하다.

3) 지속성 저장 클래스(persistent storage class)

Persistent 데이터는 장치의 전원이 끊어져도 그 값을 유지하는 특성을 가진다. MIDP에서 Persistent 데이터를 지원하는 클래스들은 javax.microedition.rms 패키지에서 정의되어 있다.

4) 그래픽 사용자 인터페이스

MIDP에서는 두 가지 GUI API가 정의되어 있

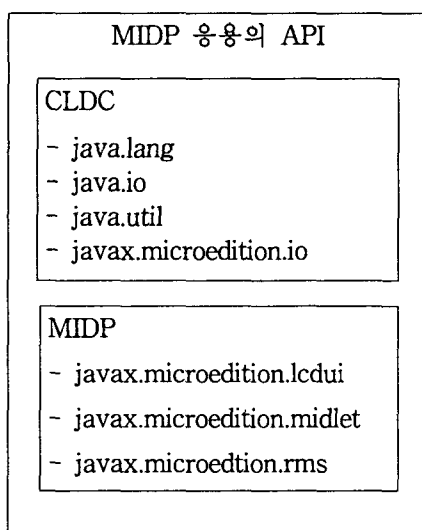


그림 5. MIDP 응용의 API

다. 하나는 장치의 이식성에 초점을 맞춘 상위 수준의 API이며, 다른 API는 특정 장치에서만 동작하는 그래픽 요소와 저수준의 이벤트 제어를 다루는 API이다. GUI와 이벤트 처리에 관한 클래스는 javax.microedition.lcdui 패키지에 포함되어 있다.

3.2 MIDlet의 라이프 사이클

MIDlet도 Java 애플릿과 비슷한 라이프 사이클을 가진다. MIDlet이 실행 중인 경우 Active, Paused, Destroyed의 상태를 가질 수 있다. MIDlet에서 정의된 startApp(), pauseApp(), destroyApp()를 실행함으로써 상태를 전환할 수 있다. MIDlet 응용 프로그램이 실행되면, AMS(Application Management Software)는 먼저 인자없는 생성자를 사용하여 MIDlet 객체를 생성하고, Paused 상태로 전환시킨다. AMS에서 startApp()를 호출하면, MIDlet의 상태는 Active로 전환되며 필요한 자원을 할당받아 실행을 시작한다. AMS에서 MIDlet이 더 이상 필요하지 않게 되거나 높은 우선 순위를 가진 프로그램을 위해 메모리가 필요하다고 판단되면, destroyApp()를 호출함으로써 MIDlet을 종료시킨다. 이후 MIDlet은 Destroy 상태로 전환되며 모든 자원을 반환하고, persistent 데이터를 저장한 후 종료된다.

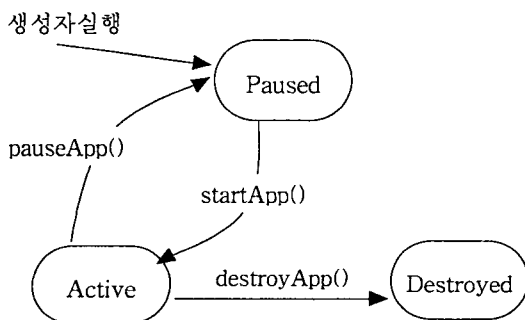


그림 6. MIDlet의 라이프 사이클

3.3 Hello World 예제 프로그램

다음은 MIDP를 준수하는 간단한 Hello World 예제 프로그램 소스이다. 그림 7은 다음 소스 코드를 실행한 결과이다.

[소스 코드]

```

import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class HelloWorld extends MIDlet implements CommandListener{

    private Display disp;
    private TextBox mainScreen = null;
    private Command exit = new Command("exit", Command.EXIT, 2);

    public HelloWorld() {
        disp = Display.getDisplay(this);
        mainScreen = new TextBox("Text Box", "Hello World", 512,0);
        mainScreen.addCommand(exit);
        mainScreen.setCommandListener(this);
    }

    public void startApp() {
        disp.setCurrent(mainScreen);
    }

    public void pauseApp() { }

    public void destroyApp(boolean unconditional) { }

    public void commandAction(Command c, Displayable s) {
        if(c == exit) {
            destroyApp(false);
            notifyDestroyed();
        }
    }
}
    
```


[실행 결과]

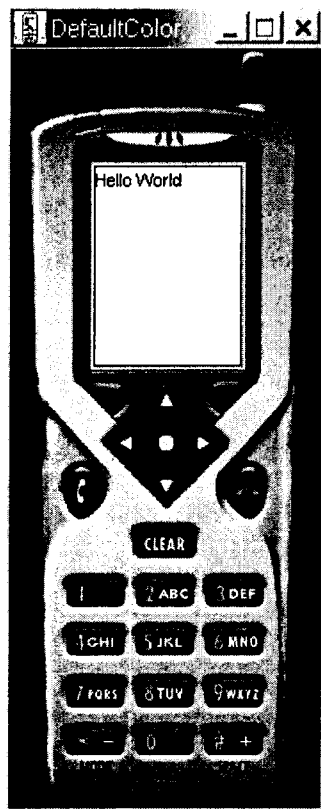


그림 7. 실행결과

4. 결론

지난 수년간 이동 통신 가입자의 수는 급격하게 증가하였으며, 2003년 말까지 전세계 가입자의 수는 10억 명에 이를 것이라는 보고서도 나와있다. 이에 따라 PDA, 휴대전화의 기능에 대한 보다 다양한 요구가 증가할 것이며, 이는 기존의 WAP로는 충족하기 어려울 것이다. WAP의 문제점을 개선하기 위해 개발된 Java ME는 분산 객체 언어로써 검증된 Java 언어를 모바일 컴퓨팅 환경에 적용한 것이다. J2ME Wireless Toolkit는 모바일 응용 프로그램을 PC에서 편집, 컴파일, 및 테스트가 가능한 에뮬레이터를 포함하고 있는 편

독점하지 않으며, 외부 전문가 그룹인 JCP(Java Community Process)의 기술 자문과 요구를 수용하며 개발하고 있다. 전세계의 많은 휴대용 기기가 자바기반으로 출시되고 있으며, 이를 이용한 멀티미디어 서비스 개발에도 더욱 확대될 것으로 기대된다.

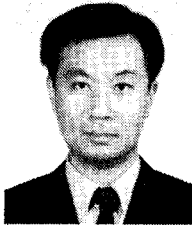
참고 문헌

1. Sun사 "Java ME 홈페이지", <http://java.sun.com/j2me/>
2. Yu Feng, Zun Zhu, "Wireless Java Programming with J2ME", Sams Publishing. 2001.
3. Jonathan Knudsen, "Wireless Java: Developing with Java 2, Micro Edition", APRESS. 2001.
4. Eric Giguere, "Java 2 Micro Edition", Wiley. 2000
5. 주예찬, 김대성, 박정수, 감명훈, 최종필, "자바 모바일 실전 프로그래밍", 정보문화사. 2001
6. H. M. Deitel, P. J. Deitel, S. E. Santry, "Advanced Java 2 PLATFORM-How to Program", DEITEL. 2001.
7. Sun microsystems, "Java2 Platform, Micro Edition Wireless Toolkit User's Guide", October, 2001.



김 일 민

- 1984 경북대학교 전자공학과 (학사)
- 1989 뉴저지 공과대학 전산과 (석사)
- 1995 아리조나 주립대 전산과 (박사)
- 1985~87 전자통신 연구원(ETRI) 연구원
- 1996~97 삼성 SDS 교육개발센터
- 1997.3~현재 한성대 컴퓨터공학과 교수
- 관심분야 : CORBA, 분산처리, 원격 교육



조 세 흥

- 1996. 아리조나 주립대 전산학(석사)
- 1999. 아리조나 주립대 전산학(박사)
- 1999~2002 대구대학교 정보통신공학부 교수
- 2002. 3~한성대학교 컴퓨터 공학부 교수
- 관심분야: 멀티미디어, 원격 교육, 가상대학



박 재 희

- 1984 경북대학교 전자공학과(학사)
- 1992 Texas A&M 전자공학과(석사)
- 1995 Texas A&M 전자공학과(박사)
- 1984~90 국방과학연구소(연구원)
- 1995~97 삼성전기 주식회사(부장)
- 1997. 3~현재 계명대학교 교수
- 관심분야: 음성보안, 메일 보안, 컴퓨터 통신