

## 모바일 자바를 이용한 공학용 계산기 개발

김 성 호\*

### Development of the Engineering Calculator using Mobile Java

Sung-Ho Kim\*

#### 요 약

본 논문에서는 사용자가 원하는 수식을 휴대 전화기의 사용자 입력 인터페이스를 통해 한꺼번에 모두 입력하면, 입력받은 수식을 분석하고 계산 및 연산자 우선순위를 파악한 후에 순차적으로 계산을 수행해주는 휴대 전화기용 공학 계산기 개발에 대하여 기술한다. 기존의 계산 기능을 지원하는 휴대 전화기에서는 수식 전체를 한꺼번에 입력 받지 못하는 단점으로 인하여 간단한 수식 계산을 수행할 때에도 여러 번의 수식 입력을 해야 하기 때문에 하나의 수식을 계산하는데 시간이 많이 소모되는 불편함을 가지고 있었다. 또한 입력받은 수식을 저장할 메모리의 부족과 수식 분석 기능 등의 부재로 인해 현재까지 휴대 전화기용 공학 계산 기능은 제대로 서비스 되지 못하고 있다. 그러므로 본 논문에서는 이러한 문제점들을 개선하여 모든 수식을 한꺼번에 입력하고 계산 및 연산자 우선순위에 의하여 실시간으로 결과 값을 확인할 수 있는 시스템을 개발하였으며, 향후 공학용 계산기가 필요한 학생 및 일반인들에게 매우 유용하게 사용될 수 있을 것으로 기대한다.

▶ Keyword : Mobile, Engineering Calculator, Java Virtual Machine, Numerical Formula Analyzer, Algorithm

---

• 제1저자 : 김성호  
\* 상지대학교 컴퓨터정보공학부

## 1. 서론

인간이 간편하게 휴대할 수 있는 휴대 전화기가 10여 년 전에 처음으로 일반인들에게 선보이기 시작하였다. 그리고 그 이후부터 점차 휴대 전화기가 대중화되면서 최근에는 거의 모든 사람들이 휴대 전화를 사용하고 있다는 것을 모르는 사람은 거의 없다. 이러한 현상이 나타남에 따라 휴대 전화기와 관련된 다양한 연구 분야가 발전하고 있으며, 휴대 전화기에서 지원되는 기능들도 사람들의 수많은 욕구를 만족시키기 위하여 매우 다양해지고 있다. 그러나 아직까지 휴대 전화기에서 공학용 계산 기능이 제대로 수행되어지는 휴대 전화기는 찾아보기가 어려운 상황이다. 물론 최근 유통되고 있는 휴대 전화기들 중에서 계산 기능을 지원하는 휴대 전화기[2, 10]가 있기는 하지만 수식 전체를 입력 받지 못하는 단점을 가지고 있다. 즉, 계산 기능을 어느 정도 지원하는 기존의 휴대 전화기에서는 단순히 하나의 피연산자를 입력받고 그 후에 연산자를 다시 입력하여 계산 결과를 얻어내는 방식을 사용하고 있다. 만약 입력되는 수식이 끝나지 않은 경우, 이를 반복 수행하여 결과를 출력한다. 그로 인하여 간단한 수식 계산을 수행할 때에도 여러 번의 수식 입력을 해야 하기 때문에 하나의 수식을 계산하는데 소모되는 시간이 많이 드는 불편함을 가지고 있다. 이러한 문제점들이 바로 기존 휴대 전화기가 가지고 있는 수식 계산 기능의 한계이며 모바일 기기의 한계이기도 하다. 또한 입력받은 수식을 저장할 메모리의 부족[3]과 수식을 분석하는 수식 분석 기능[1, 5] 등의 부재로 인해 현재까지 휴대 전화기용 공학 계산 기능은 서비스 되지 못하고 있다. 이러한 배경 하에 본 논문에서는 처음부터 사용자가 원하는 수식을 휴대 전화기의 사용자 입력 인터페이스를 통해 한꺼번에 모두 입력하도록 한다. 수식 분석기에서는 입력받은 수식을 분석하여 계산 우선순위 및 연산자 우선순위를 파악한 후에 순차적으로 계산을 수행[1, 5]함으로써 기존의 휴대 전화기용 수식 계산 기능의 단점을 보완하고자 한다.

본 논문의 구성 및 각 장의 내용은 다음과 같다. 2장은 휴대 전화기용 공학 계산기 시스템의 설계 내용을 소개하며, 3장에서는 수식 입력의 핵심 기술인 수식 분석 기능 알고리즘에 대하여 기술한다. 4장에서는 공학 계산기 구현 결과를 보여 주고, 마지막 5장에서는 결론 및 향후연구를 언급하고 끝낸다.

## II. 공학용 계산기 시스템 설계

휴대 전화기용 공학 계산기를 설계하기 위해서는 먼저 휴대 전화기에서 자바 버추얼 머신(Java Virtual Machine, JVM)[9]의 동작 원리와 기본 구조의 지식이 필요하다.

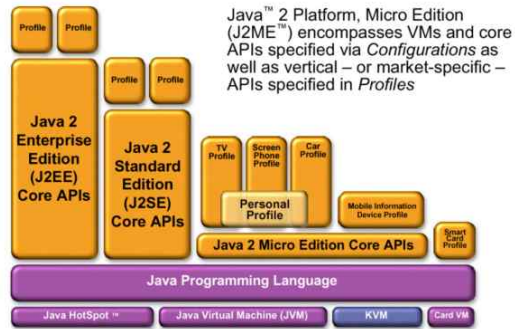


그림 1. Java™ 2 Platform [11]

[그림 1]은 Java™ 2 Platform으로서 모바일 기기의 자바 가상 머신 플랫폼(JVM Platform)을 간단하게 확인할 수 있다. 지금부터는 Java™ 2 Platform을 기반으로 공학용 계산기 설계에 필요한 플랫폼 및 API[4, 6, 7]를 살펴보고 구현 내용에 대하여 간단히 기술하도록 한다.

### 2.1 J2ME (Java 2 Platform Micro Edition)

J2ME는 이동 전화 및 개인 정보 단말기(PDA, Personal Digital Assistants)와 같은 이동 무선 정보 기기용 프로그램을 개발하기 위한 자바 2 프로그래밍 언어 버전이다. 그리고 SUN Microsystems에서 개발한 가상 머신이며 특수 자바 번역기인 KVM(K Virtual Machine)을 사용하며, CLDC(Connected, Limited Device Configuration)와 MIDP(Mobile Information Device Profile)에 기반을 둔 무선 응용 프로그램 개발용 도구[12]이다. 또한 제한된 내장형 기기 환경에서 동작하는 자바 프로그램을 개발하기 위한 플랫폼으로, 휴대 전화기나 PDA, 가전제품 등과 같이 일반 PC보다 저용량의 메모리, 저속의 CPU 등 제한된 환경에서 주로 사용한다. 이러한 환경에서 사용되는 애플리케이션을 개발하기 위해서는 적은 크기의 VM(Virtual Machine)과 기본적인 자바 API보다 많이 축소된 API가 사용된다.

### 2.2 SK-VM

SK-VM(SK-Virtual Machine)은 (주)SK텔레콤이 (주)XCCE와 공동 개발한 무선 인터넷 서비스 플랫폼[9]이다. J2ME의

KVM과 동일한 기능을 제공하는 XVM(eXtended Virtual Machine)을 사용하며, CLDC를 구현한 M-Configuration과 MIDP를 구현한 M-Profile을 탑재했을 뿐만 아니라 (주)SK텔레콤이 자체 개발한 API인 SKT-Service API를 탑재하고 있다. SK-VM은 WAP(Wireless Application Protocol)와 WML(Wireless Markup Language)을 사용하여 서버에 접속하는 방식이다. 이것은 실제로 SK 통신사업자의 휴대 전화기에 해당 프로그램을 다운 받을 수 있도록 지원하며, 이것의 API를 사용하여 프로그램을 구현해본 결과, 개인의 휴대 전화기에서 다운로드 및 실행을 확인할 수 있었다.

### 2.3 SK-VM API : MathFP

MathFP[8] 클래스는 부동 소수점 연산을 클래스 레벨로 지원한다. CLDC에서는 float나 double형을 지원하지 않지만, MathFP에서는 long 및 double형을 지원하고 있다. 그런 이유로 본 논문에서는 공학용 계산기를 구현하는데 큰 어려움이 없었다. 지수 함수나 로그 함수 등과 같은 수학 함수는 자바 API에서 제공하는 method를 이용해야 한다. 따라서 SK-VM API에서 제공하는 MathFP 클래스를 이용하여 구현하였으며, 사용한 API는 [표 1]과 같다. [표 1]에서는 실험에 사용한 함수의 종류와 그에 대응되는 API를 보여주는 데, 지수 함수는 static long exp(long 타입형 변수)로 입력하여 사용된 변수의 멱지수를 계산하여 준다. 그리고 로그 함수는 static long log(long 변수)로 입력을 받아 변수의 자연로그를 계산하여 주며, 삼각 함수에서는 sin, cos, tan 등 세 가지가 필요한데, 각각의 변수를 입력으로 받아 그에 대응되는 값을 계산하여 주며, 주로 원형좌표를 직각좌표로 변환할 때 사용한다. 제곱 함수와 제곱근 함수는 지수 함수, 로그 함수 및 삼각 함수 등과 동일하게 변수를 입력받아 각각 그것의 제곱 값과 루트( $\sqrt{\quad}$ )의 계산결과를 출력하여 준다.

표 1. 실험에 사용한 SK-VM API(MathFP)의 종류

함수 종류	SK-VM API : MathFP
지수 함수	static long exp()
로그 함수	static long log()
삼각 함수	static long cos() static long sin() static long tan()
제곱 함수	static long pow()
제곱근 함수	static long sqrt()

### 2.4 주요 개발 내용

본 논문에서 개발한 휴대 전화기용 공학 계산기의 기능은 [표 2]와 같이 정리되어질 수 있다.

표 2 공학용 계산기의 기능

기능	설명
공학용 계산기	사칙 연산 및 지수 로그 함수 계산
수치 적분	적분 계산
수치 미분	미분 계산
분산과 표준편차	분산과 표준편차 계산
행렬 연산	행렬의 합과 차, 곱 계산
연립 방정식	계수와 상수를 통해 결과 계산
벡터 연산	벡터 연산 기능
방정식 해구하기	다항식의 하나의 근을 구함
보간법 근사함수	스플라인 함수와 근사 직선을 통해 그 자료들의 구간 계산
수열의 합과 수열의 곱	수식입력과 변수의 처음 값 끝 값 입력을 통해 결과 계산
좌표 변환	직각좌표나 원통좌표를 입력한 뒤 다른 좌표로 변환한 결과 출력

## III. 수식 분석기를 위한 알고리즘

본 장에서는 휴대 전화기용 공학 계산기에서 수식을 한 번에 모두 입력받고, 입력받은 수식의 계산 우선순위와 연산자의 우선순위를 분석하는 수식 분석기에 대하여 기술한다. 수식분석기의 핵심 내용은 연산자의 우선순위에 따른 단항식의 계산과 제한자(괄호 연산자)가 사용된 수식에서 계산 우선순위에 따른 다항식의 계산이다. 여기서 단항식과 다항식의 계산에 대하여 간략히 설명하면 다음과 같다.

여러 항을 포함하는 수식은 각 항을 동시에 계산할 수 없으므로 순서에 따라 계산을 수행한다. 제한자(괄호 연산자)로 둘러싸인 영역은 블록이라 하는데, 물론 블록은 또 다른 블록을 포함할 수 있다. 층 번호를 부여하는 규정은 제일 바깥 블록을 1층으로 설정하고 안쪽으로 갈수록 2층, 3층 등으로 부르기로 하는데, (식 1)을 사용하여 간단히 살펴보도록 하자.

$$2 * (\sin(\text{PI}/2) + \cos(\text{PI})) \dots\dots\dots (\text{식 } 1)$$

|-----| |-----| -> 3층  
|-----| -> 2층  
|-----| -> 1층

(식 1)의 계산 우선순위는 층 번호가 가장 높은 3층부터 시작하여 2층, 마지막으로 1층을 계산한다. 이와 같이 단층식의 계산은 제한자(괄호 연산자)를 사용하지 않고, 단 한 번에 계산하는 방식을 의미하며, 다층식의 계산은 제한자(괄호 연산자)를 수 개 이상씩 사용하여 여러 층을 형성하면서 연산자 우선순위에 의하여 계산하는 방식을 의미한다.

수식의 구조는 제한자(괄호 연산자)와 연산자의 층 번호에 의해 결정되며, 층 번호 배열은 제한자(괄호 연산자)와 연산자의 층 번호를 나타낸 배열이다. 수식 스트링에서 제한자(괄호 연산자) 및 연산자의 위치를 배열로 나타낸 것이 바로 '제한자(괄호 연산자) 및 연산자의 위치배열'이라 한다. 연산자의 연산순서와 제한자(괄호 연산자) 코드를 나타낸 배열이 '제한자(괄호 연산자) 및 연산자의 코드배열'이다. 위의 세 배열을 결합해서 하나의 이중 배열로 나타낸 것이 수식의 구조 배열이다.

표 3. 연산자 우선순위 코드

제한자 및 연산자	(	)	^	/	*	-	+
연산자 코드	-1	0	1	2	3	4	5

[표 3]은 연산자의 우선순위 코드를 보여주며, 연산자의 우선순위는 연산자 각각의 코드가 가장 낮은 것이 우선된다. 연산자 코드 중 '-1'이 제일 우선으로 계산이 수행되고 그 뒤로 '0', '1', '2', '3', '4' 등이 계산되며 연산자 코드 '5'가 가장 마지막에 수행되도록 설계하였다. 연산자 '\*', '/', 또는 연산자 '+', '-' 는 수학에서도 우선순위가 동일시되어 개발자가 임의로 순서를 설정해도 되는데, 본 시스템에서는 연산자 '/', '-' 을 좀 더 높은 우선순위로 처리하였다. 지금부터는 단층식과 다층식의 계산을 위한 알고리즘에 대하여 기술하도록 한다.

### 3.1 단층식의 계산 알고리즘

단층식의 계산을 위해서는 먼저 수식의 연산자 코드와 그 연산자 코드 배열을 연산 순서대로 재배열한 그 연산자의 위치(즉 연산자 코드 배열에서의 인덱스)와 함께 2중 배열을 만들어 step 배열을 만든다. 그 후 연산자 수를 계산하여 그 항

의 수보다 길이가 하나씩 작아지는 부분 배열을 갖는 2중 배열을 생성하여 이를 피연산자 배열로 삼는다. 참고로 모든 수식에서 피연산자 수는 연산자 수보다 1개 더 많다. 그런 다음 피연산자 배열 첫 번째 인덱스(opnd[0])에 수식의 각 항을 대입한다. 첫 번째 연산에 해당하는 연산 배열의 첫 번째 인덱스(step[0]) 연산을 수행하여 그 결과를 피연산자 배열의 두 번째 인덱스(opnd[1])에 대입한다. 같은 방식으로 다음 순위의 연산 배열의 첫 번째 인덱스(step[1]) 연산을 수행하여 그 결과를 피연산자 배열의 세 번째 인덱스(opnd[2])에 대입한다. 이와 같은 방식으로 모든 연산이 수행될 때까지 반복하면 사용자가 원하는 계산 결과를 얻을 수 있다.

### 3.2 다층식의 계산 알고리즘

다층식의 계산은 연산자의 층 번호와 연산자 코드에 의해서 계산 순서가 결정된다. 이와 같은 다층식의 계산을 위한 계산 순서 규칙을 살펴보면 다음과 같다.

첫째, 층 번호가 높은 것을 우선적으로 수행하며, 둘째, 동일 층에서는 연산 코드가 낮은 것, 즉 연산 순서가 빠른 것을 우선으로 처리한다. 셋째, 동일 층 및 동일 연산 코드일 경우에는 왼쪽 연산자를 우선적으로 처리한다. 넷째, 연산 코드가 0일 경우에는 아무런 계산 없이 내려온다.

이와 같은 계산 순서 규칙에 따라 단층식의 경우처럼 step 배열을 만든다. 그런 다음 단층식과 마찬가지로 피연산자 배열을 생성한다. 마찬가지로 수식 계산을 수행함에 있어서 피연산자가 하나씩 줄어들므로 길이가 하나씩 작아지는 부분 배열을 갖는 2중 배열을 생성한다. 피연산자 배열의 첫 번째 인덱스 opnd[0]은 제한자(괄호 연산자)와 연산자들 사이의 부분 스트링으로 채워준다. 예를 들어,

$$\sin(\text{PI}) + \text{sqrt}(1+2.99) \dots\dots\dots (\text{식 } 2)$$

(식 2)의 피연산자는 opnd[0] = {sin, PI, sqrt, 1, 2.99}와 같이 저장되는데, step[0] 연산부터 수행하며 성분을 하나씩 줄여나가는 과정으로 최종 결과를 구한다. 이와 같은 다층식의 계산 알고리즘을 정리하면 다음과 같다.

- 단계 1) 수식 구조 배열을 구한다.
- 단계 2) 수식 구조 배열로부터 연산 배열을 구한다.
- 단계 3) 피연산자 배열을 생성한다.
- 단계 4) 피연산자 배열의 첫 번째 인덱스(opnd[0])를 구한다.
- 단계 5) i=0 단계에서 i=st. length-1 단계까지 계산을 수행한다

단계 6) 최종 계산 결과는  
`result = opndst.length()`이 된다.

### 3.3 층 번호 결정 알고리즘

층 번호 결정 알고리즘이란 연산자와 제한자의 층 번호를 결정하는 것으로 다음과 같이 정리할 수 있다.  
 최초의 수식 요소는 무조건 층 번호가 1인데, 임의의 수식 요소가 오른 제한자(')일 때, 층 번호는 다음과 같다.

- ▶ 왼쪽 수식 요소가 왼 제한자('(')이면, 왼쪽 수식 요소의 층 번호와 같다.
- ▶ 왼쪽 수식 요소가 오른 제한자(')')이거나 연산자이면, 왼쪽 수식 요소의 층 번호 보다 1이 더 작다.

반대로 임의의 수식 요소가 왼 제한자('(')이거나 연산자 일 때, 층 번호는 다음과 같다.

- ▶ 왼쪽 수식 요소가 왼 제한자('(')이면, 왼쪽 수식 요소의 층 번호보다 1 크다.
- ▶ 왼쪽 수식 요소가 오른 제한자(')')이거나 연산자이면, 왼쪽 수식 요소의 층 번호와 같다.

## IV. 공학용 계산기 구현 결과

공학용 계산기를 구현하는데 있어서 매우 중요한 것 중의 하나는 수식 분석기인데, 사용자가 수식을 입력하면 수식 분석기 클래스에서 분석된 수식 스트링을 실제 계산을 수행하는 수식 계산기 클래스로 넘기게 된다. 그러면 계산기 클래스에서 `apply(operand1, operand2, 'op')` 형식의 method와 `apply(argument, "function")` 형식의 method를 이용해서 실제로 계산을 수행한다. 그리고 계산된 결과 값을 결과 출력을 담당하는 method로 넘긴다. [그림 2]는 공학용 계산기의 작동 흐름도를 표현한 것이다.

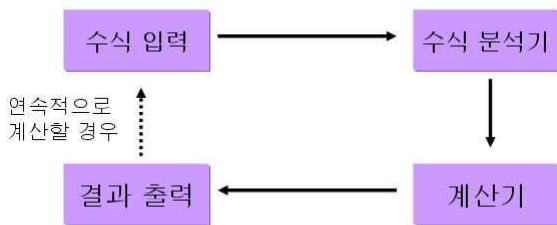


그림 2. 계산기 작동 흐름도

공학용 계산기의 작동 과정은 [그림 2]의 첫 번째 단계인 수식 입력과 같이 사용자가 휴대 전화기의 입력창에 수식을 입력하는 것으로 시작한다. 그리고 입력된 수식 스트링은 수식 분석기에서 수식 계산 우선순위와 연산자 우선순위에 따른 연산을 수행하도록 분석되어진다. 또한 분석된 수식 스트링은 계산을 한 후, 결과 값을 휴대 전화기의 출력 창에 보여준다. [그림 3]은 본 논문에서 개발한 휴대 전화기용 공학 계산기를 실행한 결과의 예시 화면이다. 휴대 전화기에 공학용 계산기 프로그램을 다운로드 받은 후 실행을 하면, [그림 3]의 '(a) 실행초기 화면'과 같이 휴대 전화기의 LCD(Liquid Crystal Display) 창에 프로그램 명칭이 간단하게 나타난다. 그리고 메뉴 버튼을 눌러주면 [그림 3]의 '(b)메인 메뉴'와 같이 공학용 계산기의 주요 기능 목록을 보여준다.

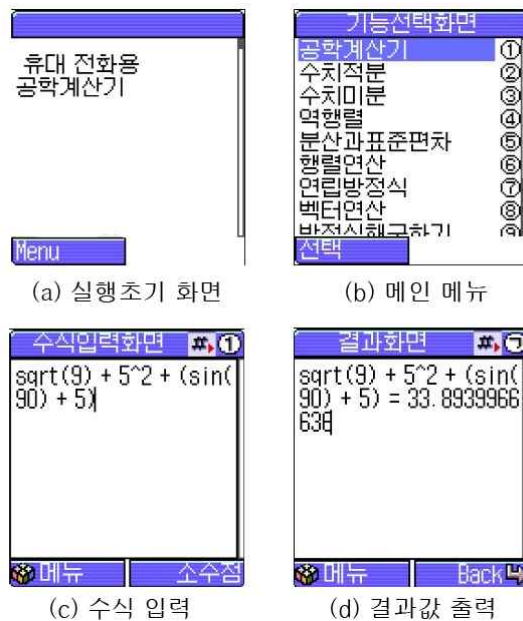


그림 3. 휴대 전화기에서 시험한 공학용 계산기

메인 메뉴에서는 본 논문에서 구현한 모든 수식 계산과 관련된 기능을 선택할 수 있다. 만약 사용자가 메인 메뉴들 중 '공학계산기' 기능을 선택하면 수식 입력 화면이 [그림 3]의 '(c)수식 입력'과 같이 나타나게 된다. 사용자는 여기서 원하는 수식을 입력하고, 수식 입력이 완료되면 메뉴에서 '계산 결과 출력'이라는 명령을 통해 [그림 3]의 '(d)결과값 출력'과 같이 결과 값을 바로 확인할 수 있다. 이와 같이 본 논문에서 구현한 공학용 계산기는 기존의 계산기와는 다르게 모든 기능

들이 연속된 수식을 한꺼번에 입력하면 바로 결과를 확인할 수 있다는 장점이 있다.

## V. 결 론

우리는 개인적으로 공학용 계산기를 사용하여 다양한 수식들을 계산하여 왔었다. 그러나 기존의 공학용 계산기는 경제력이 거의 없는 학생들이 소지하기에는 고가인데다 불편하기 때문에 누구나 쉽게 가지고 다닐 수 있는 제품은 아니었다. 그런 이유로 최근에는 거의 모든 사람들이 가지고 있는 휴대 전화기에서 수식 계산을 쉽게 할 수 있도록 하기 위하여 계산기 프로그램들이 연구 개발되어져 왔다. 그러나 기존에 개발된 계산기들은 연산자 수행 부분에서 불편함이 있어왔다. 본 시스템은 이러한 문제점들을 해결할 수 있도록 하였으며, 일반 휴대 전화기에 본 시스템을 설치한다면 휴대 전화기 만으로도 간단한 공학 수식 계산을 매우 쉽게 수행할 수 있다는 장점을 가지고 있다. 그러나 최신 휴대 전화기의 특성상 일부 복잡한 계산을 수행할 경우에는 연산 속도가 다소 느리다는 단점도 함께 가지고 있다. 그러므로 본 시스템이 보완해야 할 점은 현재 시중에 나와 있는 최신 휴대 전화기보다 낮은 버전의 SDK(Software Development Kit)로 공학용 계산기를 구현하는 것이다. 그러나 만약 그렇게 할 경우 최신 휴대 전화기에서 공학용 계산기가 실행되지 않을 수 있다는 문제점이 다시 발생한다. 그러므로 차후에 개발사에서 일반인에게 최신 SDK가 공개된다면, 최신 휴대 전화기의 버전에 맞게 보완할 수 있을 것이다. 그리고 본 논문을 통해 개발된 휴대 전화기용 공학 계산기의 다소 느린 연산 속도 부분의 문제점을 보완한 다음 최신 휴대 전화기에 이식이 된다면, 실생활에서 학생 및 일반인들에게 많은 도움이 될 것으로 기대한다.

## 참고문헌

- [1] 이규봉, "실습하며 배우는 수치해석학", 경문사, 2003.
- [2] 강성운, 이경범, 홍성인, "자바 모바일 프로그래밍", 대림출판사, 2002.
- [3] 임동혁, "게임을 만들며 배우는 자바 프로그래밍", 제우미디어, 2002.
- [4] 김대성, 주예찬, "자바 모바일 실전 프로그래밍", 정보문화사, 2001.
- [5] Steven C. Chapra, Raymond P. Canale, "공학도를 위한 수치해석 5판", Mc Graw Hill, 2006.

- [6] 김성환, 양석호, "모바일 자바 프로그래밍", Prentice Hall, 2005.
- [7] 케이시 시에라, 버트 베이츠, "Head First Java", 한빛미디어, 2003.
- [8] XCE 개발자(developer)들의 공간, <http://www.developerzone.co.kr/>.
- [9] 강상원, "모바일 플랫폼 천하통일 위피 프로그래밍", 제우미디어, 2004.
- [10] 김현철, 김보라, 공기석, 서대영, "WIPI 기반 모바일 공학용 계산기", 한국정보과학회 2007 가을 학술발표 논문집 제34권 제2호(D), pp.281~285, 2007.
- [11] Mobile Java Developer Community, <http://www.mobilejava.co.kr/>
- [12] 네이버 용어사전, <http://terms.naver.com>