

수학과 수와 연산 영역의 클래스 구조 설계

김민정⁰, 김갑수
서울용산초등학교⁰, 교육대학교 컴퓨터교육과
yulban@paran.com⁰, kskim@snue.ac.kr

Class Hierarchy of Number and Operations in Mathematics

Min-Jeong Kim⁰, Kap-Su Kim
Dept. of Computer Education, Seoul National University of Education

요 약

초등학교 수학과와 학습은 학습자의 구체적인 조작을 통해서 이루어져야 하나 현재 웹을 통하여 학습할 수 있는 수와 연산 영역의 학습 프로그램은 대부분 단순한 수의 나열이거나 프로그램에서 제공하는 학습 내용을 따라해 보는 수준에 그치고 있다. 구체적인 조작을 할 수 있는 자바 애플릿을 제작하려고 해도 초등학교의 인지 단계 및 교육과정에 적합한 수와 연산 클래스가 체계화되어 있지 않아 클래스의 사용에 어려움이 있다. 이에 본 연구에서는 수와 연산 영역의 교육과정을 분석하여 객체를 정의하고, 객체의 속성과 메소드를 분석하여 클래스를 구성한 후 클래스 사이의 관계를 파악하여 클래스 계층구조를 설계하였다. 1)

1. 서론

1.1 연구의 필요성 및 목적

초등학교 수학과와 학습은 초등학교의 인지 수준에 따라 주로 생활 주변 현상이나 구체적 사실을 학습 소재로 구체적 조작 활동을 통하여 원리나 법칙을 학생 스스로 발견하고 해결할 수 있는 기회를 제공하고자 한다. 따라서 초등학교들이 수와 연산의 여러 개념과 원리를 제대로 이해하기 위해서 구체적인 조작이나 시각적인 표상이 제공되어야 한다. 그러나 실물로서 구체물을 제공하는 것은 현실적으로 많은 제약이 따르므로 컴퓨터 프로그램을 고려해 볼 수 있다. 컴퓨터가 가지고 있는 그래픽, 시뮬레이션 등의 기능과 신속한 처리 능력, 접근의 용이성, 조작의 편리 등은 실물을 통한 조작 활동에서의 비효율성을 줄이고 역동적으로 수학 개념을 탐구할 수 있는 학습 환경을 제공해 줄 수 있다. 또한 컴퓨터 프로그램의 시각적인 효과로 학습에 대한 흥미를 높일 수 있고 학습자 중심의 개별 학습을 가능

하게 한다. 특히 수학과와 연산 영역의 경우 각 단계별로 수의 종류와 범위 제한을 통하여 초등학교 수학의 전 단계를 다룰 수 있다[1].

따라서 컴퓨터를 활용한 수학 학습에 대한 다양한 연구가 이루어지고 있으며 다양한 교육용 소프트웨어도 개발되고 있다. 최근에는 웹을 기반으로 한 코스웨어의 개발이 주를 이루고 있으며 플래시를 이용하여 교수 학습 자료를 많이 제작하고 있다. 그러나 CD-ROM으로 제작된 프로그램은 학습자가 쉽게 구할 수 없고 웹상의 접근이 가능한 코스웨어나 멀티미디어 교수 학습 자료는 특정 학습 주제에 국한되어 있거나 학습자가 구체물을 조작하여 그 변화를 통하여 개념과 원리를 이해하기 어려운 경우가 대부분이다.

자바 애플릿은 학습자가 웹을 통해 쉽게 접근하여 모델화한 구체물을 통하여 수 개념과 연산의 원리를 학습할 수 있고 사용자의 입력과 그에 따른 반응을 통해 실시간 상호작용이 가능하다는 장점을 가지고 있다. 또한 소프트웨어에 대한 비용이 들지 않으며 업데이트가 용이하다. 인터넷에는 다양한 수학교육 관련 자바 애플릿 사이트가 있지만 초등학교의 인지 수준에 적합한 것은 찾아보기 힘들며, 대체

1) 《본 연구는 한국과학재단 목적기초연구(R01-2003-000-10449-0)지원으로 수행되었음.》

로 단편적인 학습 요소에 대한 애플릿을 모아둔 것에 불과하다[2].

따라서 본 연구에서는 초등학교 수학과 수와 연산 영역을 분석하여 이를 바탕으로 초등학교 학생의 인지 수준에 적합한 수와 연산 영역의 자바 클래스를 구성하고 그 계층구조를 설계하고자한다. 초등수학 수와 연산 영역의 클래스 계층구조의 정립을 바탕으로 클래스 라이브러리를 구현할 수 있고, 이는 향후에 수와 연산 영역에 관련된 자바 애플릿 및 교육용 소프트웨어를 제작할 때 구조화된 클래스 라이브러리를 활용하여 시간과 노력을 줄이는 효율적 프로그래밍을 하기 위함이다.

1.2 연구의 내용 및 방법

본 연구에서 초등학교 수학과 수와 연산 영역의 내용을 분석하여 수와 연산 클래스를 구성하고 그 계층구조를 정립하는데 목적이 있다. 세부적인 내용은 다음과 같다.

첫째, 초등학교 수학과 수와 연산 영역의 교육과정을 분석하여 수와 연산 요소와 각 요소별 학습 요소를 추출한다.

둘째, 추출한 수와 연산 요소와 학습 요소를 바탕으로 각 객체의 속성과 메소드를 정의한다.

셋째, 정의한 객체를 바탕으로 수와 연산 클래스를 구성한다.

넷째, 클래스 간의 계층구조를 설계한다[3].

1.3 연구의 제한점

본 연구는 초등학교 학생의 인지 수준에 맞는 수와 연산 학습에 필요한 기능을 갖춘 클래스를 구성하고 구현해보고자 하였다. 따라서 본 연구는 초등학교 수준에 국한하여 수와 연산 객체를 정의하고 클래스를 구성한다.

2. 이론적 배경

2.1 초등학교 수와 연산 학습의 특성

초등학교 수학교육은 중학교나 고등학교의 수학교육과 다른 점이 많이 있다. 피아제에 의

하면 초등학교 학생들의 인지 단계는 구체적·조작적 사고기이므로 아이들은 현실 상황에서 직접적으로 관찰하거나 구체물이나 반구체물을 이용한 구체적 조작 활동을 통해서만 사고가 가능하다.

지식을 구성하는 방법도 학생들에게 수학적 경험을 하게하고, 탐구하는 활동을 하게하며 문제를 조사하고 분석하여 해결하는 경험을 바탕으로 생각함으로써 이루어지도록 권장하고 있다.

초등학교 학생이 수학적 경험을 통해 지식을 구성하는 것은 주로 구체적 조작 활동을 통하여 이루어진다는 점을 감안하여, 새로운 개념이나 내용의 학습 지도 과정에 가능한 구체적 조작 활동을 해 보일 수 있거나 학생 스스로 실험해 볼 수 있는 도구를 활용하는 것은 매우 바람직하다. 따라서 기존의 시청각 기자재 외에도 수학 활동이나 수학적 사고 실험을 구체적으로 해 볼 수 있도록 고안된 교육 기자재의 활용은 저극 권장할 필요가 있다.

초등학교 수학의 학습과 지도에 활용할 수 있는 교육 기자재의 예를 들면, 십진 막대, 분수 막대, 점판, 계산기, 컴퓨터 등으로 특히 계산기는 초등학교에서 필요한 '최소한의' 계산 기능과 그 원리를 완전히 습득한 경우에는 이를 문제 해결 과정에 적절하게 활용함으로써 문제 해결 학습의 효과를 배가시킬 수 있게 된다[4].

컴퓨터는 수학의 내용과 교수 방법, 학습 방법을 바꾸어 놓고 있다. 전통적인 교수 방법에서는 교사가 가장 중요한 정보의 출처였으나, GSP와 같은 프로그램의 출현으로 모의실험이나 탐색을 통한 다양한 학습이 가능해지고 있다. 즉 개인별, 그룹별, 탐구학습을 통해 교사도 모르는 사실을 발견할 수 있게 되었다. 컴퓨터를 수학의 보조 도구로 사용함으로써 기대되는 효과는 첫째, 시뮬레이션이 가능하고 둘째, 그래픽과 애니메이션이 주는 시각적 효과가 높으며 셋째, 반복적인 모델들을 탐구하는 것을 가능하게 하며 넷째, 학생들의 능동적 참여를 기대할 수 있다[5].

2.2 수와 연산 학습 프로그램의 문제점 및 개선 방향

수와 연산 학습 프로그램 개발을 위한 클래스 개발의 필요성은 기존 수학과와 수와 연산 학습 프로그램의 분석을 통해 명확하게 설명될 수 있다. 이론적 배경을 통해 보았듯이 초등학교 수학과와 학습은 현실 상황에서 직접적으로 관찰하거나 구체물이나 반구체물을 이용한 구체적 조작 활동을 통해서만 사고가 가능하다. 수학과와 학습은 전통적 학습 환경인 확실적인 교재의 사용과 일방적인 설명식 강의로는 이해하기가 쉽지 않다. 또한 실제수업에서 활용되고 있는 구체물이나 시각적 자료는 그 시간적 공간적 제약으로 인하여 부분적인 활용에 그치고 있는 실정이다. 이러한 학습 환경에 따른 한계를 극복하기 위한 수단으로 제기되고 있는 것이 컴퓨터이다. 그래서 초등 학생들이 흥미를 가질만한 다양한 교육용 소프트웨어와 교수 학습 자료가 개발되어 왔다.

그 중에서 초등 수학 교육에 해당하는 프로그램은 주로 Flash를 이용한 웹 저작물이나 CD-ROM 자료가 많았다. 본 연구와 같이 자바 언어로 제작된 프로그램은 주로 중등 수학 교육 위주였으며 체계성이나 통합성이 없이 단순히 애플릿을 항목별로 탑재해 놓은 것에 불과했다. 초등 수학 교육에 해당하는 프로그램의 경우에는 대체로 단편적인 학습 요소에 대한 애플릿으로 구성되어 있었고 그것도 한 두 가지 주제에 대한 애플릿만이 탑재되어 있었다.

Flash로 된 학습 프로그램의 경우는 굉장히 다양한 종류의 프로그램이 있었지만 대부분 교수 학습 단계별로 교과서의 학습 활동 순서를 그대로 옮겨 놓았거나 활동의 일부분만을 간단한 조작해 볼 수 있는 경우가 많았다. CD-ROM으로 제작된 프로그램은 학습자의 입장에서 쉽게 구하기 힘들다는 단점이 있으며, Flash와 CD-ROM으로 구현된 자료는 학습자가 직접 구체적인 조작을 통해 자발적으로 개념을 형성하기가 어렵고 매 실행시마다 같은 수의 연산이 반복적으로 이루어지게 되

어 암기식 학습이 될 우려가 있다. 또한 수와 연산 영역의 계통성을 살리지 못하고 단편적인 학습 단계별 학습으로 이루어져 있다. 따라서 학습자가 구체적인 조작을 통해 개념을 형성할 수 있고 수학의 계통성을 살릴 수 있는 프로그램의 개발이 필요한 실정이다.

2.3 객체 지향 설계의 개념

객체 지향 기술은 클래스와 객체를 기반으로 한 새로운 프로그래밍 접근 방법으로 다음과 같은 특징을 갖고 있다.

첫째, 객체지향 설계는 개발자로 하여금 이전보다 훨씬 실세계에 가까운 모델링을 할 수 있도록 해준다.

둘째, 객체지향 설계를 이용하면 프로토타입을 빠르게 구성할 수 있다. 객체지향 설계는 추상화(abstraction)가 가능하기 때문에 개발자는 객체지향 프로그램을 빠르게 만들고 수정할 수 있다.

셋째, 객체지향 설계를 이용하면 코드의 재사용(reuse)이 가능하다. 만들어진 객체는 설계과정을 다시 거칠 필요 없이, 이후에 또 다른 애플리케이션에서 쉽게 사용될 수 있다.

넷째, 객체지향 설계는 개발자가 급속히 변화하는 환경에 맞춰 작업할 수 있도록 도와준다. 객체지향 프로그램은 실세계의 변화와 같이 쉽고 빠르게 변경이 가능하기 때문이다[6].

2.4 객체와 클래스

객체 지향 프로그래밍(OOP)에서는 모든 것을 '객체'로 표현한다. 객체는 그 상태를 나타내는 자료 부분과 행동 양식을 나타내는 연산 부분으로 표현될 수 있다. OOP에서 클래스는 특정 종류의 객체 내에 있는 변수와 메소드를 정의하는 일종의 틀(template)이다. 따라서 객체는 클래스로 규정된 인스턴스로서, 변수 대신 실제값을 가진다. 클래스는 OOP를 정의하는 개념 중 하나인데, 클래스에 대한 중요한 몇 가지의 개념들은 다음과 같다.

첫째, 클래스는 전부 혹은 일부를 그 클래스 특성으로부터 상속받는 서브클래스를 가질

수 있으며, 클래스는 각 서브클래스에 대해 수퍼클래스가 된다.

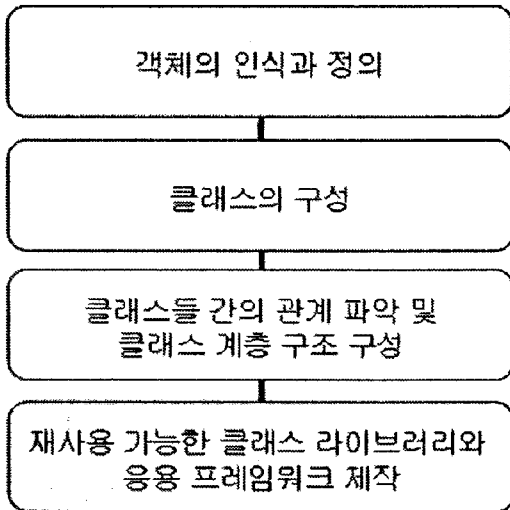
둘째, 서브클래스는 자신만의 메소드와 변수를 정의할 수도 있다.

셋째, 클래스와 그 서브클래스 간의 구조를 "클래스 계층(hierarchy)"이라 한다[7].

2.5 객체지향 소프트웨어 설계 방법론

객체지향 설계는 <그림 1>과 같은 네 가지 기본 단계를 거친다.

<그림 1> 객체 지향 설계의 네 가지 단계



객체지향 설계의 결과는 클래스의 계층구조이다. 각 클래스는 제어와 데이터 구조가 함께 포함된 모듈이다. 문제 공간(Problem Space)은 객체와 그것에 관련된 메소드들의 모임으로 좀더 자연스럽게, 사실적으로 관찰될 수 있다.

객체는 객체지향 설계의 초기 요소이다. 그 후로 점차 객체들 간의 공통점이 인식되면서, 같은 공통점을 갖은 객체들로 클래스를 만들게 되고, 또한 이들 클래스로부터 좀더 추상적인 추상 클래스(Abstract Class)를 형성하게 된다. 이와 같은 추상화 작업의 가장 상위 단계는 프레임워크(Framework)이다. 프레임워크는 서로 연관된 응용의 모임을 위한 설계를 나타내는 클래스들의 집합이다. 객체 지향 설계는 순환적이다. 프로그래머는 클래스의 집합에서 설계를 시작해서, 그것을 확장시키고 고쳐서 하나의 응용 프로토타입으로 꾸며 맞추

다. 주로 최종 사용자와의 교류를 통하여 프로토타입이 개선되길 원하면, 설계자나 프로그래머는 다시 분석/설계 작업을 반복한다. 이러한 재작업을 통하여 클래스는 개선되어 재구성되며, 만일 이렇게 개선된 클래스가 추후의 응용에 재사용될 수 있을 만큼 일반적이라면 그것을 표준 클래스 라이브러리(Standard Class Library)에 추가시킨다. 프레임워크 설계자는 응용 프로그램간의 유사성을 발견하여, 미래의 비슷한 문제에 대한 프로그래밍 해결책으로 유용한 프레임워크를 개발한다[8].

3. 초등학교 수학과 도형 영역의 클래스 계층구조 설계

3.1 초등학교 수학과 교육과정 분석

3.1.1 수와 연산 영역 내용 체계 분석

초등학생의 인지 수준에 적합한 수와 연산 학습 프로그램을 설계를 위한 클래스를 구성하고 그 계층구조를 정립하기 위해서는 우선 초등학교 수학과 수와 연산 영역의 내용을 분석할 필요가 있다. 초등학교 수학과 수와 연산 영역을 분석함으로써 초등학생이 학습해야 할 수와 연산 학습 요소를 추출할 수 있고 이는 수와 연산 객체 인식으로 연결된다. 또한 교육과정 내용 분석을 통해 수와 연산 요소별로 필요한 학습 요소를 통합적으로 정리할 수 있는데 이것은 수와 연산 객체의 속성과 메소드 정의로 연결된다.

먼저 제 7차 초등학교 수학과 교육과정의 수와 연산 영역 중 수 영역에 해당하는 학습 내용 체계를 수의 영역에 따라 <표 1>과 같이 정리하였다.

<표1>초등학교 수학과 수 영역의 내용 체계표

영역 단계	0과 자연수	분수	소수
1-가	·0 ·50까지 수		
1-나	·100까지 수 ·수 세기		
2-가	·1000까지 수		
3-가	·10000까지 수	·분수의 이해	

3-나		·단위분수와 진분수	·소수(첫째 자리)의 이해
4-가	·다섯 자리 이상의 수	·여러 가지 분 수	
4-나		·비와 몫으로 서 분수	·소수 세 자리 수의 이해
		·분수, 소수의 크기 비교	
5-가	·약수,공약수,최대공약수 ·배수,공배수,최소공배수 ·약분과 통분		
6-가		·소수와 분수의 관계 ·소수와 분수의 크기 비교	

다음으로 제 7차 초등학교 수학과 교육과정의 수와 연산 영역 중 연산 영역에 해당하는 학습 내용 체계를 연산의 영역에 따라 <표 2>와 같이 정리하였다[9].

<표 2> 초등학교 수학과 연산 영역의 내용 체계표

영역 단계	덧셈과 뺄셈	곱셈과 나눗셈
1-가	·간단한 수의 덧셈과 뺄셈 ·덧셈과 뺄셈의 활용	
1-나	·한 자리 수의 덧셈과 뺄셈 ·두 자리 수의 덧셈과 뺄셈 (받아내림, 받아올림 없음) ·덧셈과 뺄셈의 활용	
2-가	·두 자리 수의 덧셈과 뺄셈 ·곱셈의 도입 ·덧셈과 뺄셈의 활용	
2-나	·세 자리 수 범위에서 덧셈과 뺄셈 ·덧셈과 뺄셈의 활용	·곱셈 구구 ·곱셈의 활용
3-가	·세 자리 수의 덧셈과 뺄셈	·나눗셈의 도입 ·곱셈의 나눗셈 ·곱셈과 나눗셈의 활용
3-나	·세 자리 수의 덧셈과 뺄셈	·곱셈과 나눗셈
4-가	·분모가 같은 분수의 덧셈과 뺄셈	
	·자연수의 사칙연산	
4-나	·소수의 덧셈과 뺄셈	
5-가	·분모가 다른 분수의 덧셈과 뺄셈	·분수의 곱셈
5-나		·분수, 소수의 곱셈과 나눗셈
6-나		·분수와 소수의 나눗셈

3.1.2 수의 종류에 따른 수와 연산 영역의 학습 요소 분석

초등학교 수학과 수와 연산 영역의 학습 요

소를 수의 종류에 따라 <표 3>과 같이 정리하였다. 각 단계별 학습 순서에 따르되 중복되는 부분은 제외하였다.

<표 3> 수와 연산 요소와 학습 요소

수와 연산 요소	학습 요소
0과 자연수	·0 ·50까지 수 ·100까지 수 ·수 세기 ·1000까지 수 ·10000까지 수 ·다섯 자리 이상의 수 ·약수, 공약수, 최대공약수 ·배수, 공배수, 최소공배수 ·약분과 통분 ·간단한 수의 덧셈과 뺄셈 ·한 자리 수의 덧셈과 뺄셈 ·두 자리 수의 덧셈과 뺄셈(받아내림, 받아올림 없음) ·두 자리 수의 덧셈과 뺄셈 ·곱셈의 도입 ·곱셈 구구 ·곱셈의 활용 ·세 자리 수의 덧셈과 뺄셈 ·나눗셈의 도입 ·곱셈의 나눗셈 ·곱셈과 나눗셈의 활용 ·자연수의 사칙연산
	·분수의 이해 ·단위분수와 진분수 ·여러 가지 분수 ·비와 몫으로서 분수 ·분모가 같은 분수의 덧셈과 뺄셈 ·분모가 다른 분수의 덧셈과 뺄셈 ·분수의 곱셈
	·소수(첫째 자리)의 이해 ·소수 세 자리 수의 이해 ·소수의 덧셈과 뺄셈
	·분수, 소수의 크기 비교 ·소수와 분수의 관계 ·소수와 분수의 크기 비교 ·분수, 소수의 곱셈과 나눗셈 ·분수와 소수의 나눗셈

교육과정의 학습 내용 체계를 분석한 결과 수의 종류 및 연산의 종류에 따라 각 단계에서 계통성을 이루고 있음을 알 수 있다. 다시 말해 수의 종류와 크기 등의 조건의 조절을 통해 하나의 메소드로 여러 단계의 동일 주제 학습이 가능하도록 구성되어 있다. 따라서 계통성을 이루고 있는 공통 학습 주제에 따라 학습 요소를 다시 정리하되 실제 학습 상황을 생각하여 <표 4>와 같이 세부 학습 요소로 정리해보았다. 여기에서 학습의 단계는 고려하

지 않았으며, 공통적인 주제는 같이 묶어 정리하였다.

<표4>수와 연산 요소와 학습 요소

수의 연산 요소		세부 학습 요소
학습 주제	수	
약속하기 읽기/쓰기 세기	0과 자연수	·모형을 보고 수로 나타내기 ·수를 보고 문자로 나타내기 ·문자를 보고 수로 나타내기 ·1의 자리 읽기/쓰기 ·10의 자리 읽기/쓰기 ·100의 자리 읽기/쓰기 ·1000의 자리 읽기/쓰기 ·만 이상의 자리 읽기/쓰기
	분수	·모형을 보고 분수로 나타내기 ·분수 읽기/쓰기 ·분수의 종류 알기 ·자연수의 나눗셈을 분수로 나타내기
	소수	·모형을 보고 소수로 나타내기 ·소수 읽기/쓰기
순서 대소 비교	0과 자연수	·모형 이용하여 비교하기 ·1의 자리 비교하기 ·10의 자리 비교하기 ·100의 자리 비교하기 ·1000의 자리 비교하기 ·큰 수 비교하기(단위별로)
	분수	·그림으로 나타내어 비교하기 ·분자 비교하기 ·분모 비교하기 ·통분하기
	소수	·그림으로 나타내어 비교하기 ·소수 첫째 자리 비교하기 ·소수 둘째 자리 비교하기 ·소수 셋째 자리 비교하기 ·정수부 비교하기
덧셈	0과 자연수	·덧셈식으로 나타내기 ·모형 이용하여 더하기 ·1의 자리 더하기 ·10의 자리 더하기 ·100의 자리 더하기 ·1000이상의 자리별로 더하기 ·받아올리는 수 더하기
	분수	·덧셈식으로 나타내기 ·그림으로 나타내어 더하기 ·분자끼리 더하기 ·정수부끼리 더하기 ·분모 비교하여 통분하기 ·분수 종류 파악하여 가분수는 대분수로 나타내기 ·받아올림 처리하기
	소수	·덧셈식으로 나타내기 ·그림으로 나타내어 더하기 ·소수 첫째 자리 더하기 ·소수 둘째 자리 더하기 ·소수 셋째 자리 더하기 ·정수부 더하기 ·받아올림 처리하기

멜셈	0과 자연수	·멜셈식으로 나타내기 ·모형 이용하여 빼기 ·1의 자리 빼기 ·10의 자리 빼기 ·100의 자리 빼기 ·1000이상의 자리별로 빼기 ·받아내리는 수 하위 자리에 더하기
	분수	·멜셈식으로 나타내기 ·그림으로 나타내어 빼기 ·분자끼리 빼기 ·정수부끼리 빼기 ·분모 비교하여 통분하기 ·분수 종류 파악하여 가분수는 대분수로 나타내기 ·받아내림 처리하기
	소수	·멜셈식으로 나타내기 ·그림으로 나타내어 빼기 ·소수 첫째 자리끼리 빼기 ·소수 둘째 자리끼리 빼기 ·소수 셋째 자리끼리 빼기 ·정수부끼리 빼기 ·받아내림 처리하기
곱셈	0과 자연수	·곱셈식으로 나타내기 ·두 수 바꾸어 곱하기 ·피승수를 승수의 각 자리별로 곱하기 ·받아올림 처리하기
	분수	·곱셈식으로 나타내기 ·분자끼리 곱하기(자연수 포함) ·분모끼리 곱하기 ·분자와 분모의 최대공약수 찾아 기약분수로 나타내기 ·가분수는 대분수로 나타내기
	소수	·곱셈식으로 나타내기 ·소수를 자연수처럼 곱하기 ·소수점 이동하기 ·받아올림 처리하기
나눗셈	0과 자연수	·식으로 나타내기 ·똑같이 나누어보기 ·몫 알아보기 ·나눗셈 검산하기 ·피제수의 큰 자리 수부터 크기 비교하며 나누기
	분수	·분수를 자연수로 나누기 ·나누는 수를 역수로 취하여 곱셈하기
	소수	·모형 이용하여 나누기 ·소수를 자연수의 나눗셈처럼 나누기 ·소수점 위치 이동하기 ·소수를 분수로 바꾸어 나누기
기타	0과 자연수	·약수/공약수/최대공약수 ·배수/공배수/최소공배수
	분수	·통분하기 ·약분하기

3.2 수와 연산 객체와 클래스의 구성

객체는 각각의 객체가 포함하는 자료(속성)의 기술하는 부분과 객체가 나타내는 행동(메소드)을 기술하는 부분으로 표현된다. 따라서 수와 연산 객체를 정의하기 위해서는 수 개념 및 연산의 원리를 기초로 속성과 메소드를 규정해야한다.

수와 연산의 개념 정의는 다음과 같다. 수는 셀 수 있는 사물의 많고 적음, 크고 작음, 순서, 위치 등을 표시하기 위해 사용되는 기호들이다. 그리고 연산은 수, 함수 등에서 일정한 법칙에 따라 결과를 내는 조작으로 여기에서는 초등학교 수학과 교육과정에서 다루는 사칙연산을 의미한다.

따라서 수와 연산의 객체는 수와 연산의 개념 정의에 따라 수의 크기와 관련된 요소를 속성으로 가지며, 이러한 속성에 접근하고 조작할 수 있는 연산을 메소드로 가진다.

3.2.1 수와 연산 객체의 정의

객체, 수, 연산의 정의와 교육과정을 통해 분석한 수와 연산 요소를 바탕으로 초등수학 수와 연산 영역에서 규정할 수 있는 객체는 <표 5>와 같다.

<표5>초등수학 수와 연산 영역의 객체

수와 연산 요소	객체
0과 자연수	<ul style="list-style-type: none"> · 0 · 한 자리 자연수 · 두 자리 자연수 · 세 자리 자연수 · 네 자리 자연수 · 다섯 자리 이상의 자연수
분수	<ul style="list-style-type: none"> · 진분수 · 가분수 · 대분수
소수	<ul style="list-style-type: none"> · 순소수 · 혼합소수

3.2.2 도형 영역 객체의 속성 및 메소드 분석

클래스는 유사한 객체들의 공통된 특징들을 추상화한 것이다. 클래스는 그 클래스에 속한 모든 객체가 가질 속성과 실행해야 할 메소드를 정의한다. 그러므로 클래스를 만들기 위해서는 각 객체의 공통된 속성과 메소드를 분석

하는 과정이 우선되어야 한다.

<표 6>은 교육과정을 기초로 객체들의 공통되는 속성과 메소드를 분석한 것이다. 세부적인 연산은 간략하게 제시하였으며 메소드 순서는 대상이 되는 수의 개수에 따라 제시하여 점선으로 구분하였다.

<표6>객체의 공통 속성 및 메소드

객체	속성	메소드
0		· 자연수 객체 생성하기 · 0, 자연수 읽기 · 0, 자연수 쓰기
한 자리 자연수		· 자연수 모형 옮기기 · 분수로 변환하기 · 소수로 변환하기
두 자리 자연수	· 수의 길이 · 각 자리 수의 크기	· 두 자연수 크기 비교하기
세 자리 자연수	· 수 모형의 종류와 수	· 두 자연수의 덧셈 · 두 자연수의 뺄셈 · 두 자연수의 곱셈 · 두 자연수의 나눗셈
네 자리 자연수		· 여러 자연수의 순서 알기 · 세 자연수의 혼합계산
다섯 자리 이상의 자연수		
진분수		· 분수 객체 생성하기 · 분수 모형 활용하기 · 분수 읽기 · 분수 쓰기 · 소수로 변환하기
가분수	· 분자의 크기 · 분모의 크기 · 정수부의 크기 · 분수 모형의 종류	· 두 분수 크기 비교하기 · 두 분수의 덧셈 · 두 분수의 뺄셈 · 두 분수의 곱셈 · 두 분수의 나눗셈
대분수		· 여러 분수의 순서 알기 · 세 분수의 혼합계산
순소수		· 소수 객체 생성하기 · 소수 모형 활용하기 · 소수 읽기 · 소수 쓰기 · 분수로 변환하기
	· 정수부의 크기 · 각 자리 수의 크기 · 소수 모형의 종류	· 두 소수 크기 비교하기 · 두 소수의 덧셈 · 두 소수의 뺄셈 · 두 소수의 곱셈 · 두 소수의 나눗셈
혼합소수		· 여러 소수의 순서 알기 · 세 소수의 혼합계산

자연수와 분수, 분수와 소수, 자연수와 소수 사이의 연산 문제를 해결하기 위하여 각 객체에 변환 매소드를 넣었다. 이 때, 분수 또는 소수를 자연수로 변환할 경우 그 값을 소실할 수 있으므로 값을 잃지 않는 방향으로 변환할 수 있도록 작성하였다.

3.2.3 클래스 구성

<표 6>에서는 교육과정을 바탕으로 수와 연산 객체들의 공통적인 속성과 매소드를 분석하였다. 여기에서 자연수, 분수, 소수는 “모형 활용하기, 읽기, 쓰기, 크기 비교하기, 덧셈, 뺄셈, 곱셈, 나눗셈, 순서 알기, 혼합 계산”의 공통적인 매소드를 가지고 있다는 점을 확인할 수 있다. 따라서 유사한 특징을 가지고 있는 세 종류의 수를 추상화하여 클래스를 구성할 수 있다. 수와 연산 영역의 객체와 클래스, 추상클래스를 <표 7>과 같이 정리할 수 있다.

분수의 경우에는 자연수인 분자와, 자연수인 분모로 표현되므로 자연수 클래스를 상속받아 사용할 수도 있겠지만 자연수와 분수, 소수가 모두 비슷한 연산을 사용하고 그 유사한 특징을 모아 “수”라는 추상클래스를 구상하였으므로 「0과 자연수», 「분수», 「소수」 클래스 모두 「수」 클래스를 상속받는 것으로 구상하였다.

<표7>객체, 클래스, 추상 클래스의 구성

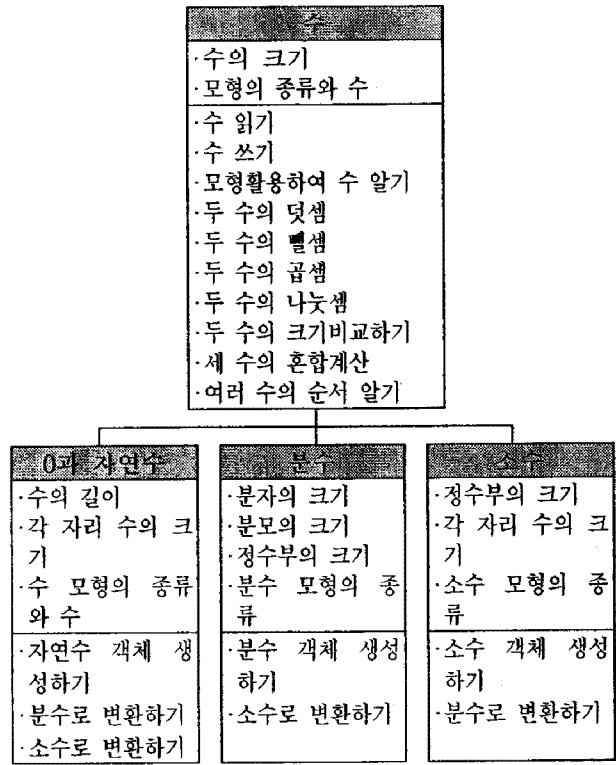
객체	클래스	추상클래스
<ul style="list-style-type: none"> 0 한 자리 자연수 두 자리 자연수 세 자리 자연수 네 자리 자연수 다섯 자리 이상의 자연수 	0과 자연수	수
<ul style="list-style-type: none"> 진분수 가분수 대분수 	분수	
<ul style="list-style-type: none"> 순소수 혼합소수 	소수	

3.2.4 클래스 계층구조

<그림 1>에서는 객체를 추상화하여 구성된

클래스의 계층구조를 나타내었다.

<그림 1>수와 연산 클래스의 계층 구조



초등학교 수학과와 모든 수의 연산에서의 공통적인 요소를 모아 「수」 클래스를 구상하였으므로 수와 연산 클래스의 계층구조에서 「수」 클래스를 최상위 클래스로 규정한다.

0과 자연수는 학습의 과정에서 함께 나타나므로 같은 클래스 내에 포함된다. 그리고 「0과 자연수», 「분수», 「소수」 클래스가 모두 「수」 클래스를 상속한다.

4. 결론 및 과제

초등학교의 수학 학습은 초등학생의 인지수준에 따라 구체적인 조작을 바탕으로 이루어진다. 자바 애플릿은 수와 연산 영역 뿐 아니라 전 영역의 수학 학습에서 구체적인 조작을 통한 학습을 가능하게 하여 효율적인 학습을 이끌어낼 수 있으나, 현재 자바 애플릿으로 구성된 학습 자료가 많지 않은 실정이다. 그러나 공개된 자바 애플릿들은 일부 단편적인 학습 주제만을 다루고 있는 경우가 대부분이며 수와 연산 영역 자바 애플릿을 제작하려고 해도 초등학교 수학 교육에 적합한 클래스 설계가

되어 있지 않아 제작에 어려움이 있다. 그러므로 초등수학 도형 교육과정에 기초한 클래스 계층구조의 정립이 필요하다.

본 연구에서는 초등학교 수학과 수와 연산 영역의 자바를 기반으로 한 클래스 계층구조를 정립하기 위하여 수와 연산 영역의 교육과정에서 학습 요소를 분석하고, 분석된 학습 요소를 통하여 객체를 정의하였다. 그리고 이러한 객체의 속성과 메소드를 분석하여 클래스를 구성한 후 클래스 사이의 관계를 파악하여 클래스 계층구조를 정립하였다. 이러한 클래스 계층구조의 정립은 그것을 바탕으로 구현된 클래스 라이브러리는 수와 연산에 관련된 자바 애플릿 및 교육용 소프트웨어를 제작할 때 코드의 재사용으로 프로그램 개발의 시간과 노력을 단축시킬 수 있다는 점에서 의미를 갖는다.

본 연구에 대한 향후 과제는 다음과 같다.

첫째, 수와 연산 영역의 클래스 계층구조를 바탕으로 실제적으로 클래스를 구현하여 클래스 라이브러리를 만든다.

둘째, 구현된 클래스 라이브러리를 활용하여 수와 연산 학습 애플릿을 개발하고, 전통적인 도형 학습과 비교하여 그 교육적 효과를 검증한다.

5. 참고문헌

- [1] 조한혁·최경식·김민정, “인터넷 상의 조작도구를 이용한 수학교육 프로그램 개발”, 韓國數學教育學會誌 시리즈 E <數學教育 論文集>, 제 13집, p549-562, 2002.
- [2] 노현정, “초등학교 수학과 평면도형 영역의 자바 클래스와 애플릿 개발”, 석사학위논문, 서울교육대학교 교육대학원, 2005
- [3] 이민희·김갑수, “초등학교 수학과 도형 영역의 클래스 계층구조”, 한국정보교육학회 1997년 하계 학술발표논문집, 제 8권, 제 1호, p187-194, 2003
- [4] 교육부 고시 계 1997-15호(1999), “초등학교 교육과정 해설(IV)”, 교육부
- [5] 임해경·강순자, “초등학교 수학교육에 있어서 컴퓨터의 활용을 위한 연구”, 韓國數學教育學會誌 시리즈 C <初等數學教育>, 제 3권, 제 1호, p47-54, 1999.
- [6] Murray Cantor, “Object-Oriented Project Management with UML”, Rational Rose, 1998.
- [7] 심재철 편역, “Thinking in Java”, 인터버전, 2002.
- [8] 장성우, “현재의 소프트웨어 관련 기술을 주도하고 있는 객체 지향 기술의 소개”, ORACLE MAGAZINE, Vol. 11 summer, pp.40-49, 1997.
- [9] 배종수, “제 7차 교육과정을 중심으로 초등수학교육 내용지도법”, 서울:경문사, 1999.