

초등학생의 논리적 사고력 향상을 위한 객체지향 프로그래밍 교육에 관한 연구

홍태진*, 박경모**

*가톨릭대학교 교육대학원 컴퓨터교육과, 오정초등학교

**가톨릭대학교 컴퓨터정보공학부, 교육대학원 컴퓨터교육과

e-mail: magister80@naver.com , kpark@catholic.ac.kr

A Study on Object-Oriented Programming Education for Improving Logical Thinking Ability of Elementary School Students

Tae-Jin Hong*, Kyeongmo Park**,

*Dept of Computer Science Education, The Catholic University of Korea

**Dept of Computer Science Education, The Catholic University of Korea

요 약

컴퓨터 프로그래밍 학습은 추상적인 개념을 이해하고 주어진 문제를 스스로 해결하도록 한다. 기존 초등학교 프로그래밍 교육에서 BASIC, C와 같은 절차적 프로그래밍 언어에 대한 연구는 많이 있지만 JAVA와 같은 객체지향 프로그래밍 언어를 통한 교육은 찾아보기 힘들다. 본 논문에서는 구조 중립적이며, 분산 인터넷 환경에 적합한 특성을 가지고 있는 객체지향 JAVA 프로그래밍 학습 시스템을 개발하여, 초등학교 학생들로 하여금 활용하도록 하였다. JAVA 학습 시스템 구현을 통한 객체지향 프로그래밍 교육은 초등학생들의 논리적 사고력을 향상시키고, 수학, 과학 과목의 학습 성취도에 긍정적인 영향을 준다는 것을 현장 학습 실험으로 확인하였다. 더불어 컴퓨터에 대한 흥미도가 상승하였다.

1. 서 론

컴퓨터 교육의 목적은 학생들로 하여금 미래사회의 중요한 도구인 정보기술 능력을 길러 주는 것이라고 볼 수 있다. 우리나라는 1960년대 후반에 컴퓨터 교육의 필요성이 제기되어, 제 3차 교육과정부터 일반 보통 교육으로써 컴퓨터 교육의 계기를 마련하게 되었으며, 제 7차 교육과정과 함께, 2000년 8월 초에 정보통신기술(ICT: Information Communication Technology) 교육 운영지침[1,2]을 제시하여 초등학교 1학년부터 정보통신기술 교육을 주당 한 시간 실시를 필수화하였다. 제 7차 컴퓨터 교육 과정은 컴퓨터 교육의 체계성이 결여되어 내용의 중복이 심하고, 컴퓨터 소양교육에 초점이 맞추어져 학생들의 문제 해결 능력 신장을 하지 못한다는 문제점을 보이고 있다.

컴퓨터를 활용하여 주어진 문제에 대한 해답을 얻는 과정은 1단계 시스템 분석, 2단계 입, 출력 결정, 3단계 알고리즘 설계, 4단계 코딩, 5단계 실행 및 테스트로 다섯 단계 [6]를 거친다. 3단계와 4단계는 컴퓨터 프로그래밍의 구체적인 과정으로, 3단계는 추상적인 개념을 갖게 되는 과정이고, 4단계는 추상적인 개념을 구체적으로 표현하는 과정이다. 컴퓨터 프로그래밍은 학습자가 컴퓨터를 활용하여 자신이 원하는 것을 수행하도록 하는 작업이다. 컴퓨터 프로그래밍을 학습하는 과정은 자연스럽게 학생들이 컴퓨터를 이해하고, 학생들의 문제 해결 능력을 신장시킬 것이다. 이는 곧 7차 교육과정에서 요구하는 자기 주도적 학습을 실현하는데 기여할 것이다.

본 논문에서는 초등학교 6학년 학생을 대상으로 객체지향 프로그래밍 언어 중 JAVA를 통하여 객체지향 프로그래밍 교육을 실시하였다. 이를 통하여 객체지향 프로그래밍 교육이 아동들의 논리적 사고력을 향상시키고, 수학과 과학 교과목의 학습 성취도에 영향을 주며, 컴퓨터에 대한 흥미를 높여 준다는 것을 확인 할 수 있었다.

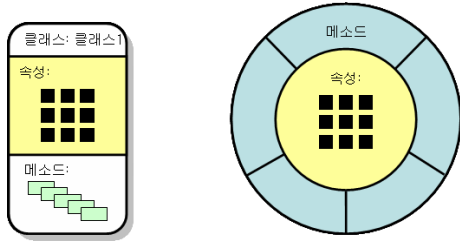
본 논문에서는 이 결과를 다음과 같은 순서로 정리한다.

2절에서는 객체지향 프로그래밍 개념에 대해서 정리하고, 3절에서는 객체지향 프로그래밍 교육 시스템에 대해서 설명하며, 4절에서는 실험대상과 실험도구, 실험설계, 실험결과와 분석을 병행하여 정리한 후 5절에서는 결론과 함께 차후 연구과제를 정리한다.

2. 객체지향 프로그래밍 개념

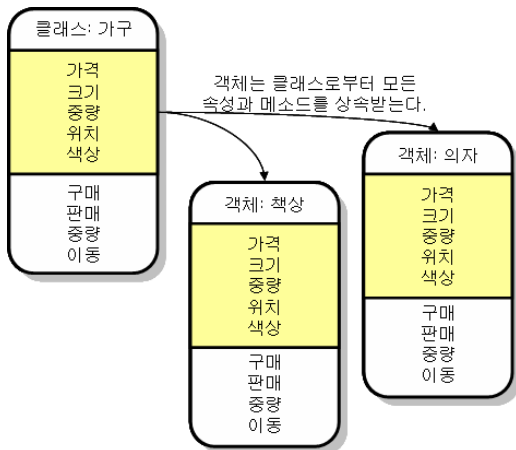
객체지향 프로그래밍(Object-Oriented Programming)의 기본 개념은 주어진 문제속의 현실세계를 실제 상황을 모형화하는 것이다. 프로그램에서 표현된 현실세계의 대상들은 그 대상이 갖는 특성을 통해 인식된다. 여기서 특성이란 그 대상을 다른 것과 구별 짓는 성질과 행동양식 또는 다른 대상과의 상호작용의 관계로 나누어진다. 따라서 객체지향 프로그래밍에서의 [그림1]처럼 클래스(class)의 인스턴스(instance)인 객체(object)는 현실세계의 시·공간상에 존재하는 개체(entity)로서 특성인 속성(attribute)과 메소드(method)에 의해 기술된다. 클래스의 속성은 메소드로 둘러싸여 있고, 속성을 동작시키기 위한 방법은 메소드 중의 하나를 통하는 것이다. 그러므로 클래스는 속성과 메소드를 캡슐화(encapsulation)한다. 캡슐화를 통하여 정보 은닉(information hiding)을 달성하며, 변경에 연관된 부작용의 영향을 감소시켜 준다. 다시 말해 클래스란 객체를 정의하는데에 필요한 모든 변수의 리스트를 작성한 것으로 하나 이상의 유사한 객체들을 공통된 속성으로 표현한 것을 말한다.

객체지향 프로그래밍에서 객체들은 서로 비슷한 부류로 나누어진다. 각 객체들은 어떠한 큰 집단에 속하거나 그 집단의 부분적인 특징을 가진다. 따라서 객체지향에서는 비슷한 성질을 가지는 객체들을 하나의 집단으로 정의한 후 특정 객체가 그 집단의 특성을 일부 또는 모두 가진다고 정의한다.



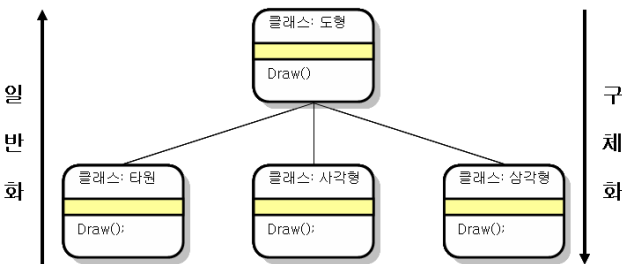
[그림 1] 객체-클래스의 표현법

[그림 2]는 클래스에서 객체로의 상속(inherit)을 나타낸 것이다. 의자는 가구라고 불리는 훨씬 큰 클래스에 속해 있는 인스턴스(instance)이다. 일반적인 속성과 메소드의 집합이 가구 클래스에 속해 있는 모든 객체에 연관될 수 있다. 모든 가구 클래스는 가격, 크기, 중량, 위치, 색상이라는 속성과 구매, 판매, 중량, 배달이라는 메소드가 있다. 의자 객체와 책상 객체는 가구 클래스에 속해 있으므로 가구 클래스가 가지고 있는 모든 속성과 메소드를 상속받는다.



[그림 2] 상속성의 개념

객체지향 프로그래밍의 개념 중 하나로 중요한 것이 바로 다형성(polymorphism)이다. 다형성은 [그림 3]과 같이 기존의 객체지향 시스템을 확장하는데 필요한 노력을 크게 줄여주는 특징이다. 예를 들어 도형을 그리는 프로그램을 생각해 보자. 도형에는 삼각형, 사각형, 타원이 있을 때, 기존에는 세 도형의 유형에 대한 모듈들을 개발할 필요가 있다. 그리고 새로운 도형을 추가하는 것이 어려울 수 있다. 새로운 모듈이 각각의 도형에 대해 생성되어야 하기 때문이다.



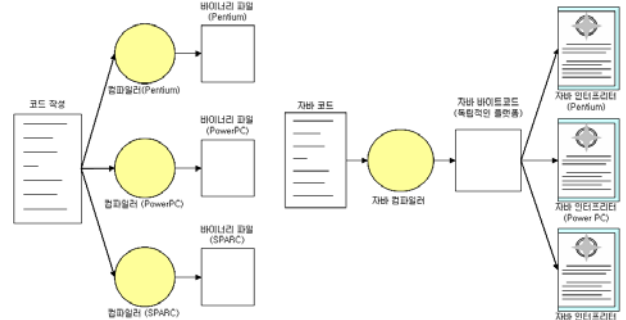
[그림 3] 메소드의 다형성

이런 문제를 해결하기 위해 클래스 타원, 삼각형, 사각형은 일반적인 도형 클래스의 서브 클래스가 되고 중복(overloading)이라는 개념을 사용하여, 각각의 서브클래스는

draw라는 메소드를 정의한다. 그러면 객체는 이들 서브클래스들의 인스턴스 객체들 어떤 것에게라도 draw 메시지를 보내고, 메시지를 받은 객체는 draw 메소드를 호출하게 된다. 새로운 도형이 추가될 때, 서브클래스는 자기 자신의 draw 메소드를 함께 생성한다. 다형성은 이처럼 많은 다른 메소드들이 같은 이름을 갖게 해준다. 이것은 객체들을 서로 분리시켜주며, 각각을 보다 독립적으로 만든다.

자바(Java)는 인터넷환경에서 널리 사용되는 객체지향형(object-oriented) 프로그래밍 언어로 썬 마이크로시스템즈(Sun Microsystems)사가 95년 말에 발표하였다. 자바는 C++ 언어처럼 보이지만, C++ 보다는 사용하기에 간단하고 프로그래밍의 객체지향성을 강화하였다. 이처럼 자바는 인터넷 분산 환경에 적합하고, 구조적으로 중립적이며, 이식성이 뛰어난 고성능의 인터프리터 언어라는 특징이 있다.

[그림 4]는 고전적인 컴파일 방법과 자바의 컴파일 방법을 비교하여 나타낸다. 좌측과 같이 사용자가 작성한 소스 코드를 실행시키기 위해서는 운영체제에 맞는 소스 코드를 작성하여, 컴파일 하여야 했다. 하지만 우측에서와 같이 자바로 작성한 코드는 자바 컴파일러라는 자바 가상머신(JVM: Java Virtual Machine)을 통하여 바이트 코드로 컴파일 하게 된다. 독립적인 플랫폼을 가지게 되는 자바 바이트코드는 어떠한 운영체제에서나 자바 가상머신만 있다면 자바 가상머신내의 자바 인터프리터를 통해 실행이 가능하다. 이점은 자바의 특징인 구조 중립적이며, 이식성이 뛰어난 고성능의 인터프리터 언어라는 것을 나타낸다. 또한 자바는 기존의 프로그램 언어의 특성을 모두 가지고 있으면서 인터넷을 환경으로 프로그램을 만들고 수행시킬 수 있는 기능도 가지고 있다.



[그림 4] 고전적인 컴파일(좌)과 자바 컴파일(우)

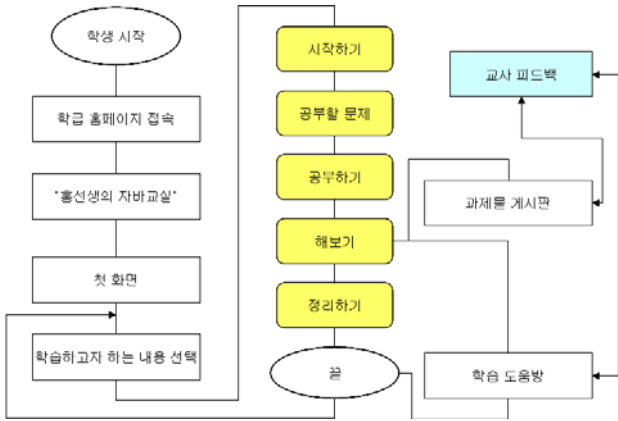
3. 객체지향 프로그래밍 교육시스템 개발

본 시스템은 Pentium 4 3.0GHz, 512MB RAM, 120GB HDD, CD-RW COMBO의 하드웨어에 Microsoft Window XP시스템 환경에서 Scanner, 540만 화소 디지털 카메라와 Macromedia Flash MX, Adobe Photoshop 7.0.1의 응용프로그램을 가지고 개발되었다. 그리고 본 시스템은 Pentium 4.0, 512MB RAM, 120GB, Microsoft Window 98 이상의 시스템 환경에서 실행가능하며, Java 2 Platform, Standard Edition 5.0 이상의 자바개발도구와 Editplus 2.31 이상의 응용프로그램을 가지고 있어야 한다.

본 시스템은 기존의 연구 중 초등학교 학생을 대상으로 한 컴퓨터 프로그래밍 교육 중 C언어나 BASIC언어에 대한 시스템은 있으나 JAVA언어에 대한 시스템은 없으며, 컴퓨터 소양교육에 치우친 현행 초등학교 컴퓨터 교육과정의 문제점을 개선하고자 시스템의 개발을 기획하였다. 본 시스템을 통하여 아동의 논리적 사고력을 향상시키고, 프로그래밍에 대한 관심과 컴퓨터 교육에 대한 흥미를 높여, 초등학교 컴퓨터 교육의 목적을 극대화시키는데 개발

의 목적을 두었다.

교육 시스템은 [그림 5]와 같이 구성되어 학습이 이루어진다. 시스템은 기본적으로 다섯 가지의 메뉴를 제공하는데 학습준비, 공부할 문제, 공부하기, 해보기, 정리하기 순으로 제공된다. 시스템의 다섯 가지 메뉴는 실제 교실 수업이 이루어지는 학습의 흐름과 같이 제공하여 학습을 하는데 있어서 학습자들이 거부감이 들지 않도록 하였다.



[그림 5] 교육 시스템 구성

교육 시스템은 [그림 7]과 같이 플래쉬를 사용하여 구현하였으며, 객체지향 프로그래밍 교육에 사용하였다.



[그림 6] 공부하기 화면

4. 실험 및 결과

본 실험은 경기도 부천시 오정구에 소재하는 오정초등학교 6학년 A학급 39명과 B학급 40명의 두 개의 집단을 임의로 선정하여, A학급에 교육 시스템을 적용하였다. 객체지향 프로그래밍 교육이 학습자에게 어떠한 영향을 끼치는가를 알아보기 위해 학습 성취도 검사지, GALT(Group Assessment of Logical Thinking) 논리적 사고력 검사지 축소판, “홍선생의 자바교실” 교육 시스템, 컴퓨터 교육에 대한 흥미도 검사 설문지를 실험도구로 삼았다.

본 실험에서 사용된 자료는 SPSS 12.0K for Windows 통계 프로그램을 사용하여 처리 되었다. 사전-사후의 논리적 사고력과 학습 성취도, 컴퓨터 교육에 대한 흥미도를 알아보기 위하여 t-test 검증을 하였으며, 연구대상의 배경 변인에 따라 각 조사 문항 간에 차이는 유의수준 1%, 5%에서 검증하였다.

4.1 논리적 사고력의 변화

교육 시스템 적용 전 실험집단과 비교집단의 논리적 사고력은 어느 정도의 동질적인 집단인가를 살펴본 사전 논리적 사고력 검사는 <표 1>와 같다.

실험 집단에 교육 시스템을 적용하기 전에 두 집단에 실시한 논리적 사고력 검사를 통해 동질성 검사를 한 결과, 실험집단의 평균점수가 4.36, 비교집단의 평균점수가 4.50으로 비교집단이 평균점수 0.14점이 높게 나타났으나, 두 집단 간 평균점수 차이에는 통계적으로 의미 있는 차이가 아니었다.

<표 1> 두 집단 간 논리적 사고력 동질성 검사

구 분	N	Mean	SD	t
실험 집단	39	4.36	2.254	-.282
비교 집단	40	4.50	2.196	

교육 시스템을 실험집단에 적용 한 후 실시한 사후 논리적 사고력 검사와 사전 논리적 사고력 검사 결과간의 차이를 알아본 결과는 <표 2>과 같다. 비교집단은 사전 논리적 사고력 검사의 사전 검사 평균 점수가 4.50에서 사후 논리적 사고력 평균 점수가 4.68로 평균점수가 0.18점 높아졌지만 유의미한 차이는 없었다. 실험집단은 사전 논리적 사고력 검사의 평균 점수가 4.36에서 교육 시스템을 적용한 후 사후 논리적 사고력 검사의 평균점수가 5.49로 평균점수가 1.13 향상되는 의미 있는 변화가 나타났다.(t=-3.561, p<.01)

<표 2> 사전-사후 논리적 사고력 검사 변화

구 분		N	Mean	SD	t
실험집단	사전	39	4.36	2.254	-6.521*
	사후	39	5.49	2.512	
비교집단	사전	40	4.50	2.196	-1.045
	사후	40	4.68	2.117	

** p<.01

4.2 학습 성취도의 변화

교육 시스템을 적용하기 전 연구대상인 실험집단과 비교집단의 사전 학습 준비도를 실시하였다. 실시한 결과는 <표 3>과 같이 두 집단은 학습 준비도에 있어서 수학, 과학 과목 모두 평균 점수에서 차이가 있었다. 하지만 두 집단 간 평균점수 차이에 대한 유의미한 차이는 없었다.

<표 3> 실험집단과 비교집단 동질성 검사

구 분		N	Mean	SD	t
수학	실험집단	39	63.08	22.844	.118
	비교집단	40	62.45	24.367	
과학	실험집단	39	78.72	15.758	-.539
	비교집단	40	80.63	15.697	
총점	실험집단	39	141.79	27.518	-.183
	비교집단	40	143.07	34.120	

교육 시스템 적용 후 사후 학습 성취도 검사를 실시한 후, 사전 검사와 비교를 해 보았다. 비교집단의 사전-사후 학습 성취도 검사 비교 결과는 <표 4>과 같다. 비교집단의 경우 각 과목마다 평균점수가 높아지거나 낮아졌으며, 총점에 있어서도 8.68점이 높아졌다. 하지만 통계적으로 의미 있는 차이라고 볼 수 없었다. 실험집단의 경우 수학 과목은

사전검사 평균 63.08에서 사후검사 평균 75.38로 12.30점이 상승한 것으로 나타났으며, 이는 의미 있는 향상이었다. ($t=-5.067, p<.01$) 과학과목의 평균은 사전검사가 78.72, 사후검사가 84.62로 5.90점이 높아졌으나 통계적으로 의미 있는 향상은 아닌 것으로 나타났다. 총점은 사전검사가 141.79에서 교육 시스템 적용 후 160.00로 18.21점이 상승하였으며, 이는 의미 있는 변화였다. ($t=4.928, p<.01$)

<표 4> 사전-사후 학습 성취도 검사 비교

구 분		N	Mean	SD	t	
비교 집단	수학	사전	40	62.45	24.367	-1.718
		사후	40	70.50	15.013	
	과학	사전	40	80.63	15.697	-.175
		사후	40	81.25	17.237	
	총점	사전	40	143.07	34.120	-1.246
		사후	40	151.75	23.082	
실험 집단	수학	사전	39	63.08	22.844	-5.067**
		사후	39	75.38	14.929	
	과학	사전	39	78.72	15.758	-1.522
		사후	39	84.62	18.510	
	총점	사전	39	141.79	27.518	-4.928**
		사후	39	160.00	31.098	

** $p<.01$

4.3 컴퓨터 교육에 대한 흥미도

교육 시스템 적용 후 실시한 컴퓨터 교육에 대한 흥미도 조사결과는 <표 5>과 같다. 컴퓨터 프로그래밍 영역의 경우 사전-사후 조사 결과 평균점수가 2.32에서 3.41로 1.09가 상승하였으며, 이는 의미 있는 변화였다. ($t=-5.702, p<.01$)

<표 5> 컴퓨터 프로그래밍 영역 흥미도 조사

구 분	N	Mean	SD	t	
실험 집단	사전	39	2.32	.941	-5.702**
	사후	39	3.41	1.125	

** $p<.01$

본 실험결과와 같이 객체지향 프로그래밍 교육 시스템의 적용을 통해 학생들의 컴퓨터에 대한 흥미도가 증가하였다는 것을 알 수 있다. 시스템의 적용 전에는 컴퓨터에 대하여 잘 모르고 있었지만, 시스템 적용 후 컴퓨터의 작동원리를 자세히 배우고 프로그래밍이라는 것을 알게 됨으로써 흥미도의 향상에 도움을 주는 것이라고 볼 수 있다. 그리고 시스템을 통하여 학습을 하면서 프로그램에 대해서 더 깊이 알고 그것과 관련된 직업이나 일에도 관심을 더 가지게 되었다고 볼 수 있다.

5. 결론

컴퓨터 교육은 컴퓨터 소양능력을 기르는 것에서 벗어나 컴퓨터를 활용하여 문제를 해결할 수 있는 능력까지 기를 수 있어야 한다. 컴퓨터를 활용하여 문제를 해결할 수 있는 능력을 기르기 위한 방법으로 컴퓨터 프로그래밍 교육이 그 대안이 될 수 있다. 따라서 본 논문에서는 객체지향 프로그래밍 개념과 자바 언어를 초등학교 학생들에게 교육 시킬 수 있는 교육 시스템을 설계하고 구현하였다.

개발된 시스템을 경기도 부천시 오정구에 소재하는 오정 초등학교 6학년 학생 79명을 실험집단(39명)과 비교집단(40

명)으로 구분하여 활용하였다. 실험 결과 교육 시스템을 적용한 실험집단이 비교집단 보다 논리적 사고력과 학습 성취도에 있어 의미 있는 향상이 나타났다. 과목별로 살펴보면, 수학 과목의 학업 성취도 향상은 의미 있는 향상을 보였으나, 과학 과목의 경우는 학업 성취도에서 많은 점수의 변화를 가져오긴 하였지만 통계적으로 의미 있는 변화는 아니었다. 그리고 교육 시스템 적용 후 컴퓨터에 관한 흥미도가 높아짐을 설문지를 통하여 알 수 있었다.

향후 연구과제는 다음과 같다. 첫째, 컴퓨터 프로그래밍 교육을 시킴에 있어서 자바와 C언어 중 어떤 것이 효과적인가에 대한 연구가 필요하다. 현재 실시되고 있는 정보 올림피아드 경진대회에서는 C/C++언어가 주로 사용되고 있으며, 프로그래밍을 배우는 학생들의 경우에 있어서도 C언어를 먼저 배운다. 자바언어와 C언어 중 어떤 것을 처음 배우는 것이 효과적인가에 대한 연구는 아직 없다. 객체지향 프로그래밍 언어와 절차형 프로그래밍 언어 중 어떤 것을 처음 배웠을 때, 그리고 배웠을 때 더 효과적이고 더 발전적인 방향을 제시하는지에 대한 비교는 흥미 있을 것이다.

둘째, 초등학교 학생들에게 컴퓨터 프로그래밍 교육은 아직도 낮은 교육내용이다. 초등교육 뿐 아니라 중등교육에서도 프로그래밍 언어에 대해서 배울 기회가 적다. 컴퓨터 프로그래밍 교육이 학생들의 논리적 사고력의 향상에 의미 있는 향상을 가져온다는 결과를 보았을 때, 컴퓨터 프로그래밍 교육에 대한 보다 많은 콘텐츠의 개발이 필요하다.

마지막으로 양질의 컴퓨터 교육을 위하여 교사들의 컴퓨터 활용능력을 높이는 것이 필요하다. 교육현장에 있는 일선 교사들 중에는 아직도 컴퓨터의 활용에 있어서 미숙한 교사들이 많다. 이 점은 컴퓨터 교육 내용을 획일화 시키고, 컴퓨터 교육에 대한 흥미를 떨어뜨리는 결과를 가져온다. 따라서 교사들을 위한 효과적으로 컴퓨터를 활용할 수 있도록 재교육 콘텐츠 개발도 필요하다.

참 고 문 헌

- [1] 교육인적자원부. 초·중등학교 정보통신기술교육 운영지침. 서울: 교육인적자원부, 2000.
- [2] 교육인적자원부. 초·중등학교 정보통신기술교육 운영지침 해설서. 서울: 교육인적자원부, 2006.
- [3] 김승현. 열혈강의 Java Programming. 프리렉, 2006.
- [4] 우치수, 강병욱, 강병도, 이명재. 소프트웨어 공학: 실무적 접근. 서울: 한국맥그로힐, 2003.
- [5] 이원규. 유현창. 김현철. 정순영. 컴퓨터 교육론. 서울: 흥릉과학출판사, 2003.
- [6] 정인정. 알고리즘. 서울: 흥릉과학출판사, 1999.
- [7] Lemay, Laura. Perkins, Charles L. Teach yourself JAVA™ 1.1 in 21Days Second Editions. Indiana: Sams.net, 1997.