

초·중등학교에서 수학교육체계와 연계된 컴퓨터 프로그래밍 교육과정과 교수방법

Computer Programming Curriculum and Teaching Method in Connection with Mathematics Education System in the Elementary and Secondary Schools

박영미*, 유관희**

충북 옥천중학교*, 충북대학교 컴퓨터교육과**

Young-Mi Park(pym1201@hanmail.net)*, Kwan-Hee Yoo(khyoo@chungbuk.ac.kr)**

요약

제 7차 교육과정에서 초·중등학교의 컴퓨터교육과정이 컴퓨터 활용 중심 내용으로 편성되어 컴퓨터를 이용한 다양한 분야의 문제를 해결할 수 있는 능력을 가르치는데 한계가 있다. 최근 들어 초·중등학교에서 미래 교육이 요구하는 창의적이며 논리적인 사고력과 문제해결력을 신장할 수 있도록 컴퓨터 교육 과정을 변화시키는 연구들이 진행되고 있다. 이러한 능력을 신장시키기 위해 필수적인 교과목의 내용이 컴퓨터 프로그래밍이지만 아직까지 학습자의 학습 동기와 학습 전이를 높일 수 있는 체계화된 프로그래밍 교육과정에 대한 연구가 미흡하다.

본 논문에서는 초·중등학교에서 컴퓨터 교육과 밀접한 관련이 있으면서 교과 내용의 연계성을 갖추고 있는 수학교육과정 내용을 분석하여 프로그래밍 관련 요소를 추출한 후, 이를 기반으로 초·중등학교에서 연속적이고 계열적 내용에 따라 컴퓨터 프로그래밍을 체계적으로 가르칠 수 있도록 프로그래밍 교육 과정을 제시한다. 그리고 본 논문에서 제안한 컴퓨터 프로그래밍 교과과정을 이용하여 프로그래밍 기법을 가르치기 위한 수업 모형과 학습지도안을 개발한다.

■ 중심어 : | 컴퓨터교육 | 컴퓨터 프로그래밍 | 프로그래밍 교육과정 |

Abstract

In the 7th education curriculum, computer education curriculum in the elementary and secondary schools is composited into the contents for the use of computers so that there are some limitations in teaching students the abilities for solving various problems of several areas using computers. Recently, the research has done to change the computer education curriculum for enhancing creativity and problem solving ability required by the future education. The contents of the main subject for enhancing them is of computer programming, however, there was not enough research on systematic programming education curriculum for leading to motivating learners and enhanced knowledge transfer to those learners.

In this paper, we analysis the contents mathematics education curriculum with consecutive contents and in tight connection with computer education and then extract its programming related elements. Based on those, we propose a programming education curriculum with which we can teach systematically computer programming according to continual and systematic guidance in the elementary and secondary schools. And we develop a teaching model and learning guidance for teaching students programming methods with the computer programming education curriculum proposed in this paper.

■ keyword : | Computer Education | Computer Programming | Programming Curriculum |

* 본 논문은 2007학년도 충북대학교 학술연구지원사업의 연구비지원에 의하여 연구되었습니다.

접수번호 : #071026-003

심사완료일 : 2007년 12월 28일

접수일자 : 2007년 10월 26일

교신저자 : 유관희, e-mail : khyoo@chungbuk.ac.kr

I. 서론

컴퓨터 교육이 사회적으로 요구되어 1988년 고시된 제5차 교육과정에서 처음으로 컴퓨터과목이 독립교과로 도입되었다[1]. 그리고 컴퓨터 교과가 선택과목이지만 독립교과로 자리잡기 시작한지 십여년이 지난 지금 컴퓨터 교육의 필요성에 관하여 의문이 제기되기 시작하였다. 이런 의문이 발생한 근원은 컴퓨터 교육이 과학적인 사고와 논리적인 문제해결 능력을 길러주고, 미래의 국가경쟁력을 담보할 우수한 과학기술 인력을 길러내는 “컴퓨터과학” 교육이 이루어지지 못하고 있다. 그리고 지나치게 컴퓨터 소양교육으로 치우쳐, 실생활에서 사용하기 불편함이 없을 정도로 교육이 이루어져 더 이상 필요하지 않다는 것이다[2]. 우리나라를 비롯한 세계 여러 나라가 정보 시대 또는 지식 기반 사회를 맞아 지식의 이해나 암기보다는 비판적 사고, 창의적 사고, 의사결정, 문제 해결, 협동 능력을 길러줄 수 있는 교육이 요구되고 있다[1]. 이러한 시대적 변화에 적응해가기 위해 컴퓨터 교육도 커다란 변화를 요구받고 있다. 이에 부응하기 위해 컴퓨터를 이용하여 실생활과 밀접한 관련이 있는 문제를 해결해가는 체계를 익힘으로써 논리적 사고방식과 문제해결 능력을 길러, 미래의 국가경쟁력을 육성할 수 있는 인재를 기를 수 있는 프로그래밍을 포함한 컴퓨터과학 교육과정과 내용이 제시되어야 하고, 이 요구가 잘 반영된 교육과정이 현장에 적용되어야 한다[3].

컴퓨터 교과의 프로그래밍 교육은 창의적 사고와 수준 높은 문제해결력을 동원하여 새로운 알고리즘을 생성해내는 정보교육의 핵심 분야[4]임에도 불구하고, 제7차 컴퓨터 교육과정에서 누락되어 있다. 또한 프로그래밍 교육 및 지도 방법에 관련된 연구가 미흡한 실정이다. 이제는 프로그래밍 교육이 학습자의 논리적 사고력, 문제해결력을 기르기 위해서는 학습자의 내적 동기를 부여하고 학습자 수준이나 관심을 고려한 학습방법을 제공할 수 있는 연구가 필요하다[5]. 따라서 본 연구에서는 프로그래밍과 문제해결 능력 신장간의 관계에 관하여 살펴보고, 프로그래밍 교육 내용을 학습자의 인지 능력에 맞추어 계열성을 가지도록 컴퓨터 교육에서

프로그래밍 교육 과정 및 내용을 제시하고자 한다. 이를 위해 초중등학교에서 컴퓨터 교육과 밀접한 관련이 있으면서 교과 내용의 연계성을 갖추고 있는 수학교육 내용 체계를 분석하였다. 또한 학습자의 문제해결력을 신장할 수 있는 수업 지도 방법을 개발하고자 한다.

본 논문의 구성은 II절에서는 관련 연구를 살펴보고, III절에서는 초중등학교에서 컴퓨터 교육과 밀접한 관련이 있으면서 교과 내용의 연계성을 갖추고 있는 수학교육과정 내용을 분석한 후, 이를 기반으로 초중등학교에서 연속적이고 계열적 내용에 따라 컴퓨터 프로그래밍을 체계적으로 가르칠 수 있도록 프로그래밍 교육 과정을 제시한다. IV절에서는 III절에서 제시된 교과과정을 이용하여 프로그래밍 기법을 가르치기 위한 수업 모형을 제시한다. 마지막으로 결론에서는 본 논문의 의의와 향후 연구방향을 언급한다.

II. 관련 연구

1. 문제해결과 프로그래밍 학습 관계

많은 교육자들은 문제해결과 컴퓨터 프로그래밍과는 깊은 관련이 있다고 믿는다. 프로그래밍과정은 문제해결 과정의 모든 면을 연습하는 창조적이고 발명적인 활동이고, 프로그래밍 활동은 문제해결에 대한 실험적 방법을 촉진시킨다[4]. 백영균[5]은 컴퓨터 프로그래밍을 문제해결기술 이외에, 자세한 것에 관심을 기울여야 되고, 방법적 사고과정이 요구되는 지적 활동을 필요로 하고 있어 프로그래밍을 인지요구활동이라고 불렀다. 또한 프로그래밍은 주어진 문제를 해결하기 위해 계획을 세우고 오류를 고치는 것 등은 프로그래밍 언어의 구문이나 어의를 기억하였다가 재생하는 것 이상의 지적 기술을 필요로 할 뿐만 아니라, 프로그래밍 환경이 사용자 즉, 프로그래머에게 주는 즉각적인 피드백 같은 독특한, 일종의 훈련방법을 제공함으로써 그 과정에서 요구되어지는 그러한 지적 기술들을 향상시켜 준다. 다만 무엇을, 어떻게, 그리고 언제 가르쳐야 하는지는 해결되어야 할 문제라고 하였다. 박판우, 이성근[4]은 프로그래밍 교육은 문제 분석 및 해결 능력, 논

리적 사고력, 절차적 문제 해결 방식 등을 습득하는데 궁극적인 역할을 한다. 또한 프로그래밍은 컴퓨터 과학의 중요한 부분으로, 컴퓨터를 이해하는데 빼 놓을 수 없는 중요한 부분이며, 프로그래밍을 통해서 다른 응용 프로그램들을 더욱 깊이 이해하고 잘 활용할 수 있는 기초를 닦을 수 있다고 하였다. 이정모 등[6]은 프로그래밍 과정을 주어진 특정 프로그래밍 과제에 대한 지식을 동원하여 문제해결 전략을 세우고 알고리즘을 구현한 후 프로그램 언어로 부호화하는 과정으로 보았다. 또한 프로그래밍 언어 습득 그 자체보다는 코드 만들기에서 문제 분석력과 이해력, 창의력을 기르고, 코드를 이해하는 과정에서 논리적 사고력을 배양하고, 오류 검증 및 수정 작업에서 반성적 사고 능력과 같은 고등인지기술을 향상시킬 수 있다고 하였다.

프로그래밍과 문제해결력과의 관계를 살펴볼 때 컴퓨터 교육에서 사고력과 문제해결력을 기르는데 프로그래밍 학습은 유용한 교육적 도구라 할 수 있다. 또한 프로그래밍 교육은 정보화 시대가 요구하는 각 교과를 초월한 확산적 사고력과 문제해결력을 신장시킬 수 있는 교육으로 아주 중요한 자리를 차지하고 있다고 볼 수 있다.

2. 문제해결력 신장을 위한 프로그래밍 지도 방법

백영균 등[7]은 LOGO 프로그래밍을 이용하여 교사의 안내 정도에 따라 안내된 발견식 수업과 지시적 수업으로 구분하고 어떤 수업방법을 적용하여 가르쳤을 때 학생들의 문제해결력 신장시키는데 효과적인지를 규명하였다. 안내된 발견식 수업 모형은 박미숙[17]의 LOGO를 이용한 문제해결력 신장을 위한 수업 모형으로 ① 구체적 행동 ② 행동의 토론 ③ 행동의 표현 ④ 프로그래밍 ⑤ 실행 ⑥ 오류수정과 ⑦ 확장과 일반화의 7 단계와 지식적 수업 모형 ① 문제 파악 ② 문제해결의 원리 제시 ③ LOGO 문제해결의 시연 ④ 적용 및 일반화의 4 단계를 이용하여 비교 수업을 한 후 지시적 수업모형에 비해 안내된 발견식 수업모형이 문제해결력의 신장에 더 효과적이고, 컴퓨터 프로그래밍 과정 즉, 문제해결과정을 중시하고 각 과정에 필요한 인지적 요구사항을 동료들과의 토론이나 소크라테스식 질문을

통하여 해결하도록 제안을 하고 있다. 유인환[8]은 프로그래밍을 통한 문제해결력 신장을 위한 학습 내용이자 방법으로 로봇 프로그래밍 교육의 의미와 가능성에서 찾았다. 로봇 프로그래밍 수업 안을 살펴보면 ① 문제 분석 ② 로봇 설계 ③ 로봇 제작 ④ 프로그래밍 ⑤ 로봇 실행 ⑥ 오류 수정의 6단계로 설계하였다. 그리고 로봇 프로그래밍을 통해 프로그래밍과 이해력, 창의력을 기르고, 코드를 이해하는 과정에서 논리적 사고력을 배양하고, 오류 검증 및 수정작업을 반성적 사고 능력과 같은 고등인지기술을 향상시킬 수 있다고 하였다.

선행연구들을 통해 프로그래밍은 문제해결력을 신장할 수 있으며, 문제해결력을 신장하기 위한 프로그래밍 접근 방법을 살펴보면 Polya의 문제해결 단계[16]를 기초로 하여 문제의 이해 및 분석, 계획 수립(설계), 계획 실행(프로그래밍), 반성(오류수정)의 단계로 나아가고 있는 것을 알 수 있다.

3. 프로그래밍과 수학교육연계성

컴퓨터 교육에서 교과 간의 학습의 전이를 살펴보면, 수학교육이 가장 학습간의 전이가 많이 이루어지고 있는 것으로 밝혀졌다. 박성현[9]은 수학교육에서 컴퓨터를 이용하면 컴퓨터과학과 수학의 통합 학습을 통해 기초적인 프로그래밍에 대한 알고리즘의 이해와 내적 표상사이의 연관성이 성립되어 추상적인 수학적 개념과 실제적인 표상들의 연관성이 지식망을 형성된다. 그러므로 수학 학습 경험을 조직하는데 있어서 컴퓨터과학과 테크놀리지는 효과적인 교수-학습 환경의 핵심적인 요소가 될 것이라고 하였다. 그리고 정치봉[10]은 컴퓨터과학과 중등수학교육의 연결성과 통합을 살펴보면, 양 교과 사이의 구별이 점점 허물어지고 통합되어간다고 하였다. 왜냐하면 새로운 정보통신기술혁명 시대에 교육 수요자인 학생, 학부모 그리고 산업체의 수학 교육에 대한 요구를 반영하지 않을 수 없기 때문이다.

미국을 비롯한 선진국에서는 이미 수학 교과 교육에 컴퓨터 활용을 통합해 가고 있다. 우리나라에서도 차기 교육과정 개정시에 이러한 변화를 대비한다면 수학교사와 정보·컴퓨터 교사는 컴퓨터과학과 수학의 관계를 이해할 필요가 있다. 컴퓨터과학은 이론적 기초로서

순수 수학과 응용 수학 분야 모두를 포함하고 있다. 컴퓨터과학은 이러한 수학의 기초 위에 소프트웨어, 정보통신기술 시스템에 관한 기술 및 공학적 접근을 위한 교육과정이 덧붙여진 연구 분야라고 할 수 있다[10]. 그리고 컴퓨터를 이용하여 추상적인 수학 내용을 시각화하여 지도할 수 있고, 이러한 시각화가 학생들의 직접적인 경험과 통제를 통하여 이루어질 수 있어 수학 학습의 어려움을 완화시켜 줄 것이다[11]. 앞으로 컴퓨터 교육과정은 컴퓨터과학 교육과 수학교육의 연계를 통해서 학생들의 수학에 대한 흥미, 문제 해결과 수학적 모델링 능력을 향상시켜 실생활의 문제를 이해하고 해결전략을 세워 해결 방법을 찾는 방향으로 나아가야 할 것이다.

III. 프로그래밍 교육과정 설계

1. 프로그래밍 교육 목표설정

교과의 목표는 교과의 성격을 고려하여 진술되어야 할 것이고, 구체적인 일반 목표는 대 영역을 중심으로 기술되어야 할 것이다. 본 연구에서 프로그래밍 영역 교육 목표를 설정하기 위해 컴퓨터교과 교육 목표 중 프로그래밍에 관한 교육 목표를 종합하고 분석하여 프로그래밍 교육에 적합한 교육 목표를 제시하도록 한다. 컴퓨터 교육의 목표와 내용에 관한 연구 중에서 프로그래밍 영역에 관련된 교육 목표를 [표 1]와 같이 정리한 후 공통적인 목표를 추출하여 다음과 같이 프로그래밍 교육 목표를 제시하였다.

주어진 문제를 이해하고, 문제 해결을 위한 가장 효율적인 알고리즘을 선택하고 설계하는 방법을 습득한 후 프로그래밍을 통해 구현함으로써 문제해결력을 신장시킬 수 있도록 한다. 이를 위해 컴퓨터의 기본적인 데이터 구조와 알고리즘의 기본 개념을 이해할 수 있도록 한다. 그리고 학습자가 쉽게 이용할 수 있는 프로그램 언어를 활용하여 실생활에서 활용할 수 있는 간단한 응용 프로그램을 작성할 수 있도록 하여 알고리즘적 사고와 프로그램 작성 능력, 문제해결력을 신장시킬 수 있도록 한다.

2. 프로그래밍 교육 내용 설정

프로그래밍 교육 내용은 이미 설정된 프로그래밍 교육 목표를 달성하기에 가장 적합한 내용으로 구성되어야 한다. 앞서 제시된 교육 목표에 따른 연구자별 교육 내용을 분석하여 새로운 프로그래밍 교육 내용을 제시하고자 한다. 우선 교육 목표에 따른 연구자별 교육 내용을 다음과 같이 살펴볼 수 있다.

표 1. 선행 연구자 프로그래밍 영역 교육 목표

연구자 (년도)	영역	목 표
윤기창 (2004)	알고리즘	주어진 문제를 분석하여 문제해결을 위한 가장 적합한 자료 구조 및 효율적인 알고리즘을 선택하고 설계할 수 있는 능력을 기른다.
	프로그래밍	컴퓨터 프로그래밍의 원리와 구조를 이해하고, 프로그래밍을 통해 주어진 문제를 해결하기 위한 알고리즘을 목적에 맞도록 표현할 수 있는 능력과 논리적 사고력 및 창의력을 기를 수 있도록 한다.
문선영 (2005)	알고리즘	알고리즘의 기본적인 개념을 이해하며 생활 속의 문제를 알고리즘을 통해 단계적으로 해결할 수 있다.
박미화 (2006)	알고리즘	알고리즘의 개념을 알고 일상의 문제 해결 상황에서 알고리즘적 사고의 아이디어를 이용할 수 있다.
	프로그래밍	구조적 프로그래밍의 개념과 특성을 알고 이용할 수 있으며 객체지향 프로그래밍과의 차이점을 알 수 있다. 또한 기본적인 문법을 익혀 프로그램의 직접 작성을 통한 프로그래밍에 관련된 문제 해결을 할 수 있다.
정보통신기술교육운영지침 (2006)	알고리즘 및 프로그래밍	다양한 정보의 종류를 인식하고 효율적인 문제 해결 방법을 찾아내는 능력을 키우도록 한다. 또한 정보통신기술의 적용이 가능한 알고리즘적 사고와 프로그램 작성 능력이 신장되도록 한다.

문선영[12]은 학생들의 논리력 향상을 위해 “생활 속의 알고리즘”영역에서 알고리즘의 개념과 알고리즘의 표현방법을 알고 생활 속에서 활용될 수 있는 기본적인 문제풀이 개념을 이해할 수 있도록 하였다. 그리고 “알고리즘 표현하기” 영역에서 컴퓨터의 기본 자료 구조의 표현들을 이해하고 흐름도를 작성하는 방법을 터득하여 단계적인 문제 해결 능력을 기를 수 있도록 하였다. 윤기창[13]은 컴퓨터 프로그래밍의 원리와 구조를 이해하고, 프로그래밍을 통해 주어진 문제를 해결하기 위한 알고리즘을 목적에 맞도록 표현할 수 있는 능력과 논리적 사고력 및 창의력을 기를 수 있도록 하였다. 그러기 위해서 프로그래밍의 기초와 기본적인 자료구조에 대

해 학습하고, 실제로 프로그래밍을 실습하도록 제시하였다.

학교 컴퓨터 교육과 밀접한 관련이 있는 정보통신기술교육 운영지침[14]에서 프로그래밍과 관련된 영역이 “정보 처리의 이해” 영역으로, 이 영역에서는 초등학교에서 고등학교 1학년까지를 5단계로 나누어 학생 수준에 따라 교육해야 할 내용을 [표 2]와 같이 담고 있다.

표 2. 정보통신기술 교육운영지침의 정보처리이해[14]

단계	학년	주요 내용
1 단계	초등1,2학년	다양한 정보의 세계 재미있는 문제와 해결 방법
2 단계	초등3,4학년	숫자와 문자 정보의 표현 문제 해결 과정의 이해
3 단계	초등5,6학년	멀티미디어 정보의 표현 문제 해결 전략과 표현 프로그래밍의 이해와 기초
4 단계	중학생	알고리즘의 이해와 표현 간단한 데이터 구조 입출력 프로그래밍
5 단계	고등1학년	프로그램 제작과정의 이해 응용 소프트웨어 제작

이상의 선행 연구자들이 제시한 목표에 따른 내용을 검토한 결과 공통적으로 제시하고 있는 교육 내용은 다음과 같다. 첫째, 논리적 사고력 및 문제해결력 신장 : 생활 속에서 활용할 수 있는 기본적인 문제를 다루어 문제해결 과정을 인식하고 해결하는 능력을 기른다. 그리고 심화된 문제를 수준별로 다루어 봄으로 논리적으로 사고하고, 문제해결 전략을 설계하여 문제가 해결되기까지의 과정을 문서로 표현할 수 있다. 둘째, 프로그래밍 작성 능력 배양 : 프로그래밍의 개념과 프로그래밍 언어의 기본 사용법을 익혀, 주어진 문제를 알고리즘 설계에 따라 실제 프로그래밍에 적용하여 올바르게 동작하는지 확인할 수 있다. 셋째, 타 교과와 연계 교육 : 수학 교육에서 계산위주의 수업을 사고력 중심으로, 개념중심에서 시각화를 통해 학생들의 수학에 대한 어려움을 극복하려는 움직임을 프로그래밍 교육과 연결시켜 수학과 컴퓨터교육이 동반적인 교과간의 역할 분담이 이루어지도록 한다. 넷째, 협력 학습 태도 형성 : 실생활에서 사용되는 예를 중심으로 팀을 구성하여 주어진 문제를 부분적으로 나누어 해결한 후 합수를 통한 인수의 전승을 통해 문제를 해결하도록 하면, 동료 간

의 역할을 분담하여 수행하는 협력 과정을 통해 협력 학습 태도를 형성할 수 있다. 그리고 소프트웨어 개발 과정을 이해할 수 있다.

3. 프로그래밍 교육 세부 내용 조직

프로그래밍 교육은 프로그램과 관련된 기초적인 내용에서부터 알고리즘까지 프로그래밍의 전반적인 내용을 다룰 수 있어야 하기에 교육 내용을 조직함에 있어 알고리즘 영역과 프로그래밍 영역으로 나누어 제시하였다. 그리고 그 과정은 학년의 증가에 따라 계열성이 유지되고, 수학교과와의 연계성과 통합성을 갖출 수 있도록 하였다. 우리나라 컴퓨터 교과와 교육과정은 초·중등학교의 학습 단원 또는 학습 영역 분류상에서 계열성이 없고 학교 급간 중복성이 심하여 유사한 내용이 계속적으로 반복된다[15]. 따라서 본 연구에서는 프로그래밍 교육 내용의 조직은 선정한 교육 내용을 기반으로 컴퓨터 교육과정에서 제시한 수준별로 내용의 범위를 정하는 문제와 수준간의 내용의 계열성을 중점으로 조직하고자 초등학교 3·4학년, 초등학교 5·6학년, 중학교, 고등학교 급별로 학습자의 수준에 맞게 구성하였다. 또한 프로그래밍 교육 내용을 학교급별로 선정함에 있어서 연속성과 통합성이 강조되어야 한다. 우리나라 교육에서 프로그래밍과 유의미한 연결성과 통합성이 있고[3], 학습간의 전이가 활발히 이루어지는 교과는 수학이다[9][10]. 때문에 프로그래밍 교육을 단계적으로 제시할 때 수학교과 내용 체계표를 분석한 후 문제를 수학교과에 연계하여 제시하고자 하였다. 그러면 수학적 지식을 프로그래밍 및 알고리즘에 적절히 이용하면 실생활에 필요한 문제를 쉽게 해결할 수 있고, 교과간의 학습 전이가 더욱 활발히 이루어져 학습자의 동기 유발을 가져올 수 있을 것이다.

이미 수학교육에서는 학습자가 학습한 내용을 실생활에 활용하기 위하여, 그리고 수학의 개념과 원리를 시각화시켜서 학습자의 이해와 활용도를 높이고자 컴퓨터를 활용하는 방안에 관해서 많은 연구가 이루어졌다. 하지만, 수학교육에서 컴퓨터 프로그래밍과 관련한 교육의 문제점은 대부분의 수학교사는 컴퓨터에 대한 이해가 부족하여 지도에 한계성이 있고, 컴퓨터를 이용

한 수업에서 수학적 지식의 의미가 변질될 가능성이 있다는 등의 부정적인 견해를 가지고 있다는 점에서 컴퓨터 전공 교사의 입장에서 프로그래밍과 수학의 연결성과 통합성을 검토하고 연구하여야 한다. 이에 본 논문에서는 체계적으로 잘 편성된 수학교과 내용을 분석하여 프로그래밍 관련 요소를 추출한 후 좀 더 체계적으로 프로그래밍 교육이 가능한 교육 과정을 만들고자 한다.

먼저 초등학교의 수학교과 내용 중에서 컴퓨터 프로그래밍으로 쉽게 구현할 수 있는 내용과 알고리즘 학습과 관련 있는 단원을 추출하면 [표 3]과 같다.

표 3. 초등학교 수학교과 내용 및 프로그래밍 관련 요소

학년	단원	교육과정 분석 내용	프로그래밍 관련 요소
3	수와 연산	네 자리수의 덧셈, 뺄셈, 곱셈, 나눗셈	양의 정수형 상수 선언, 양의 정수를 이용한 사칙연산, 출력문
	도형	각과 평면도형, 평면도형의 이동, 원의 구성 요소, 원 그리기	Logo를 활용하여 기본명령어(FD, RT), 원 그리기
	문자와 식	규칙 찾기, 예상과 확인, 문제해결과정 설명하기	알고리즘
	규칙성과 함수	규칙에 따라 여러 가지 무늬 꾸미기	알고리즘
4	수와 연산	자연수의 사칙계산, 혼합 계산, 분모가 같은 분수 크기 비교 및 덧셈과 뺄셈, 소수	양의 실수형 상수 선언, 양의 실수형 상수를 이용한 사칙연산, 연산우선순위, 관계연산자, 단순IF문
	도형	각과 여러 가지 삼각형, 삼각형과 사각형의 내각의 크기, 여러 가지 사각형, 간단한 다각형과 정다각형	Logo를 활용하여 기본명령어(FD, BK, RT, LT, Repeat), 삼각형, 사각형, 간단한 다각형 그리기
	문자와 식	단순화, 다양한 문제 적절한 방법으로 해결하기, 문제해결과정설명하기	알고리즘
	규칙성과 함수	다양한 변화의 규칙을 수로 나타내기, 규칙을 추측하고 말이나 글로 표현하기	반복문, 말과 글을 이용한 알고리즘 표현
5	수와 연산	약수와 배수, 공약수와 최대공약수, 공배수와 최대공배수, 분모가 다른 분수의 크기 비교, 분수와 소수의 곱셈과 나눗셈	정수와 실수형 변수 선언, 실수를 이용한 사칙연산, 배정문, For문, IF문, 중감연산, 나머지 연산
	도형	삼각형의 합동, 선 및 점대칭이동, 평면도형(평행사변형, 삼각형, 사다리꼴)의 넓이, 직육면체와 정육면체	Logo 변수 선언, 위치 이동(PU, PD), 달음도형 만들기, 대칭 이동 그리기, 평행 이동 그리기 정수와 실수를 이용한 사칙연산
	문자와 식	문제해결의 여러 가지 방법을 비교하여 적절한 방법 선택하기	조건문(IF문), 알고리즘
	규칙성과 함수	여러 가지 이동을 이용하여 규칙적인 무늬 만들기	반복문(For문, while문), 알고리즘

6	수와 연산	소수와 분수의 크기 비교, 이상과 이하, 미만과 초과 범위 이해, 간단한 분수와 소수의 혼합계산	실수를 이용한 사칙연산
	문자와 식	문제해결의 여러 가지 방법을 비교하여 적절한 방법 선택하기	알고리즘

초등학교 3, 4학년의 “문자와 식”과 “규칙성과 함수” 단원에 주어진 문제의 규칙을 찾고, 문제해결 과정을 말이나 글을 이용하여 알고리즘을 표현함으로써 쉽고 재미있게 알고리즘을 접할 수 있도록 한다. 이때 학습자가 관심을 가질 수 있는 생활 속 문제를 이용하여 자연스럽게 알고리즘을 세워나가는 과정을 습득하도록 한다. “수의 연산”단원에서 3학년의 정수의 사칙연산 문제를 해결하기 위해 정수형 상수 및 사칙연산, 출력문을 학습한다. 그리고 4학년은 혼합계산, 분모가 같은 분수의 크기 비교 및 덧셈과 뺄셈을 수행하도록 하기 위해서 실수형 상수 및 사칙연산, 실수를 비교하기 위해 관계 연산자, 단순 IF문을 학습하도록 한다. 또한 복잡한 연산 문제를 해결할 수 있도록 연산 우선순위를 배우도록 한다. 초등학교 3, 4학년의 “도형”단원은 Logo를 이용하여 쉽게 주어진 도형을 작도할 수 있도록 기본 명령어(FD, RT, REPEAT)를 익히고 선, 내각과 외각의 개념을 이용하여 원, 삼각형, 다양한 다각형을 작도를 할 수 있도록 한다.

초등학교 5, 6학년은 이전에 배운 내용과 “문자와 식”과 “규칙성과 함수”단원을 통해 비교, 선택, 반복의 개념이 형성될 시기이므로, 실생활에서 선택, 반복 구조를 갖는 다양한 문제를 찾도록 한다. 그리고 문제를 여러 가지 방법으로 해결해 보고 적합한 방법을 선택할 수 있도록 한다. 단, 알고리즘의 구현에서 학생들에게 부담을 주지 않는 자연어로 표현하도록 함으로써 쉽고 재미있게 알고리즘을 사용하고 표현하는 방법에 대해 학습하도록 한다. 초등학교 5학년 “수와 연산”단원의 문제를 해결하기 위해 이미 습득한 명령어를 기반으로 정수 및 실수형 변수 선언 및 배정, 단순 IF문, 단순FOR문, 나머지 연산을 익혀 문제를 해결하도록 한다. 그리고 6학년 “수와 연산”단원의 문제를 해결하기 위해 관계연산자와 정수 및 실수형 변수를 이용한 사칙연산, 혼합연산, 단순 IF문을 익히도록 한다. 초등학교 5, 6학년의 “도

형"단위에서는 Logo를 이용하여 삼각형의 합동, 선 및 점 대칭이동을 실행하기 위해서 변수를 이용한 다양한 길이의 도형 그리기, 도형의 이동, 합동을 수행하기 위해 이동(PU)과 지우기(PE) 명령어를 익혀 도형의 대칭과 합동을 수행해 보도록 한다.

다음으로 중학교 수학 교육 내용에 따른 프로그래밍 관련 요소를 추출하면 [표 4]와 같다.

표 4. 중학교 수학교과 내용 및 프로그래밍 관련 요소

학년	단원	교육과정 분석 내용	프로그래밍 관련 요소
7	수와 연산	소인수분해, 최대공약수, 최소공배수, 2진수의 덧셈과 뺄셈, 정수와 유리수의 사칙 계산	IF문, For문, 다중 For문, 1차원 배열, 논리연산자, 나머지 연산, 음수를 포함한 정수의 사칙연산, 절대값 함수
	문자와 식	문자를 사용, 식의 값, 일차 방정식	입력문, 변수, 변수를 이용한 사칙연산
8	규칙성과 함수	정비례, 반비례, 함수의 개념, 순서쌍과 좌표, 함수의 그래프 및 활용	함수의 선언, 함수 사용
	문자와 식	다항식의 연산, 지수법칙, 미지수가 2개인 일차방정식과 연립일차방정식, 부등식의 활용	입력문, IF문, while문
9	수와 연산	제곱근의 사칙연산, 실수, 근호를 포함한 식 계산, 근 사값 구하기	제곱근 함수, 반올림 연산

중학교의 알고리즘 영역은 초등학교에서 습득한 순차, 선택, 반복의 표현 방법을 이용하여 "규칙성과 함수"의 정비례와 반비례를 표현하도록 한다. 그리고 "수와 연산"단원의 최대공약수와 최소공배수 문제 해결과정에서 교환 알고리즘을 익히도록 한다. 이와 함께 실생활에서 정렬, 검색, 교환 등과 관련된 문제를 찾도록 하고, 그 과정을 알고리즘으로 표현할 수 있도록 한다. 이때 문제는 간단하면서 실생활과 관련된 문제로 하고 단계적으로 난이도를 조절하여 문제해결력을 높일 수 있는 문제를 신중하게 선정하여야 한다. 중학교의 프로그래밍 영역에는 "수와 연산"단원 문제를 프로그램으로 구현할 때, 문제 풀이 과정을 알고리즘으로 설계한 후, 프로그램으로 작성하여 실행하여 문제해결 전략 및 방법을 익히도록 한다. 이 때 이미 습득한 명령어를 제외하고 1차원 배열의 선언과 활용, 다중 For문, 논리연산자, 음의 정수 및 실수형 상수의 사칙연산, 제곱근 연산, 제곱근 함수, 절대값 함수를 습득하도록 한다. 또한

"문자와 식"단원의 다항식의 연산, 일차방정식과 연립 방정식, 부등식 문제를 해결할 때 변수, 입력문과 반복문, 조건문을 이용하여 방정식과 부등식을 계산하도록 한다. 그리고 "규칙성과 함수"단위에서는 함수의 선언 및 호출을 이용하여 프로그램을 단계별 해결하고 해결 과정을 단순화 하는 과정을 익히도록 한다.

고등학교의 프로그래밍 교육과정을 구성하기 위해 우선 고등학교 수학 교육을 살펴보면, 공통수학과 수학 I 과 수학II, 이산수학, 실용수학, 미분과 적분, 확률과 통계로 필수와 선택교과로 나누어져 운영되고 있다. 이에 고등학교 수학교과에 관련해서 공통적으로 사용되는 프로그래밍과 알고리즘적 요소를 추출하면 [표 5]와 같다.

표 5. 고등학교 수학교과 내용 및 프로그래밍 관련 요소

학년	단원	교육과정 분석 내용	프로그래밍 관련 요소
10	문자와 식	다항식의 연산, 인수분해와 약수, 배수, 이차방정식, 간단한 삼차방정식, 연립이차부등식	1차원배열, 다중 For문, 논리연산자, 나머지 연산
	규칙성과 함수	함수의 그래프, 합성함수와 역함수, 이차함수의 최대, 최소 및 응용, 일반각과 호도법, 삼각함수 성질 및 활용	2차원배열, 다중 For문, Sin함수, Cos함수, Tan함수
수학 I	지수와 로그	거듭제곱과 거듭제곱근, 지수의 확장, 로그, 상용로그	제곱근함수, 로그함수, 상용로그함수
	행렬	행렬, 행렬의 연산, 역행렬, 연립 일차방정식과 행렬	2차원 배열, 다중 for문
	수열	등차수열과 등비수열, 계차수열, 수학적 귀납법, 알고리즘과 순서도	For문, 재귀함수, while문, 스택
	지수함수 로그함수	지수함수, 로그함수	지수함수 (sqrt, pow, log, log10, exp)
	순열과 조합	순열과 조합, 이항정리	랜덤함수, For문
실용수학	계산기와 컴퓨터	계산기의 활용, 간단한 프로그래밍	변수, 대입문, 출력문, 입력문, 실수연산, 조건문, 반복문, 내장함수, 배열
	경제생활	이자계산, 적립금과 할부금, 건강보험	while문
	생활통계	평균과 분산, 그래프와 표, 확률의 뜻과 활용, 기댓값, 이항분포와 정규분포, 여론조사	2차원 배열, 다중 for문, 제곱근함수, 랜덤함수
생활문제 해결	최적화 문제 해결, 생활 문제 해결, 컴퓨터를 활용한 문제 해결	알고리즘	
이산수학	선택과 배열	경우의 수, 순열, 조합, 집합의 분할, 여러 가지 배열, 비둘기집 원리	랜덤함수, while문

	그래프	그래프, 수형도, 행렬과 그래프	배열, 다중 For문
알고리즘		수의 규칙성, 이진법으로 나타낸 수, 소수의 판정과 최대공약수, 두 항사이의 관계식, 여러 가지 수열	IF문, For문, 논리연산자, 나머지연산, 재귀함수
의사결정 과최적화	그래프와 최적화		2차원배열, 다중 For문, 알고리즘

고등학교의 알고리즘 영역에서는 수학 I의 “수열”단원의 “알고리즘과 순서도” 영역과 실용수학의 “생활문제 해결”단원과 이산수학의 “의사결정과 최적화” 단원에 관련하여 알고리즘 기술 방법으로 순서도를 활용하고, 하향식 설계와 단계적 정교화를 통해 문제에 대한 단계적 해결, 해결과정의 단순화, 구조화된 디자인의 중요성을 습득하도록 한다. 고등학교에서는 수학10의 “문자와 식”단원, 수학 I의 “수열”단원과 “행렬”단원, 실용수학의 “계산기와 컴퓨터”단원, “경제생활”단원, 이산수학의 “알고리즘”단원, “의사결정과 최적화”단원의 문제를 해결하기 위해서 이전 학습한 명령어에 2차원 배열과 복합IF문, 다중For문, Switch문을 사용하여 산술연산자, 논리연산자와 관계연산자, 나머지 연산을 문제 해결 전략에 따라 사용하도록 한다. 또한 수학10의 “규칙성과 함수”단원, 수학 I의 “수열”단원과 “지수함수와 로그함수”단원, 실용수학의 “생활통계”단원, 이산수학의 “선택과 배열”단원의 문제를 해결하기 위해 앞에서 학습한 명령어와 랜덤함수, 제곱근함수, 삼각함수, 로그함수, 자연로그함수, INT함수 등 내장 함수를 이용하여 프로그램을 작성하도록 한다.

앞에서 살펴본 학교급별 프로그래밍 단원의 내용을 정리하면 [표 6]과 같다.

표 6. 학교 급별 프로그래밍 단원 교육 내용

학교급	주 제	교 육 내 용
초등 학교 3, 4 학년	알고리즘	생활 속의 규칙 찾기, 말과 글을 이용한 알고리즘 표현
	프로그래밍	Logo의 기본 명령어 익히기, 다양한 도형 그리기(삼각형, 원, 사각형, 다양한 다각형) 양의 정수와 실수형 상수 선언, 정수 및 실수형 상수를 이용한 사칙연산, 연산 우선순위, 출력문, 관계 연산자, 단순 IF문
초등 학교 5, 6 학년	알고리즘	순차, 반복, 선택 알고리즘 알기, 실생활에서 반복 및 선택구조를 갖는 문제 찾아 해결하기, 문제 해결의 여러 가지 방법을 비교하여 적절한 방법 선택하기

중 학 교	프로그래밍	Logo를 이용하여 변수에 따른 다양한 길이 도형 그리기, 삼각형의 합동, 선 및 점 대칭 이동 구현 정수와 실수형 변수 선언 및 배정, 입력문, 변수를 이용한 사칙연산, 나머지 연산, 단순 IF문, 단순 FOR문, While문
	알고리즘	정렬, 검색, 교환 등 구현하기 쉬운 문제를 알고리즘으로 표현, 문제 해결 방법과 기술(단계별 문제 해결, 해결과정 단순화)
고 동 학 교	프로그래밍	음수를 포함한 정수 및 실수의 사칙연산, 1차원 배열의 선언과 활용, 다중 For문, 복합 IF문, 논리연산자, 제곱근 함수, 절댓값 함수, 함수의 선언 및 호출
	알고리즘	문제 해결 방법과 기술(순서도 개념 및 작성법을 이용한 문제해결, 하향식 설계와 단계적 정교화를 통해 문제에 대한 단계적 해결, 해결과정의 단순화, 구조화된 디자인의 중요성을 습득)
	프로그래밍	2차원 배열 선언과 활용, Switch문, 재귀함수, 지수함수, 삼각함수, 랜덤함수

IV. 프로그래밍 수업 모형과 학습지도안

1. 프로그래밍 수업 모형

프로그래밍을 이용한 문제해결 수업 모형은 Polya의 문제해결 단계[16], 박미숙[17]의 안내된 발견식 수업모형, 그리고 백영균 등[4]의 안내된 발견식 수업모형 실험 연구를 기초로 설정하였다. Polya의 문제해결 단계 ① 문제의식 ② 계획수립 ③ 계획실행 ④ 반성의 네 단계를 제시하였는데, Polya의 문제 해결 모형을 프로그램 개발 과정으로 다시 구상하여 설정할 수 있다. 그 과정은 ① 문제 이해 ② 알고리즘적 프로시저가 어떻게 그 문제를 해결할 수 있는지 알기 ③ 알고리즘을 공식화하여 프로그램으로 표현 ④ 그 프로그램을 정확성 측면에서 평가하고, 그 프로그램이 다른 문제를 푸는데 도구로써 사용될 수 있는지 평가로 제시할 수 있다[18]. 프로그래밍 지도와 함께 문제해결 수업 모형을 제시한 박미숙[17]과 백영균, 우인상[4]의 안내된 수업 모형의 실험 연구에서는 LOGO 프로그래밍의 수업방법이 문제해결력에 미치는 효과에 관하여 Poly의 수업모형을 토대로 안내된 발견식 수업 모형을 제시하였다. 이 때 학생들이 프로그래밍 활동을 능동적으로 할 수 있는 환경을 조성하여 주되, 교사는 문제해결 과정에 따라 학습 진행 과정을 모니터링하여 학생들의 문제해결 능력을 신장시키는데 목표를 두었다. 이와 같은 안내된 발

견식 수업은 학생들이 의식적으로 지식, 정보 또는 학습 전략을 습득해 나가도록 돕고, 습득된 지식 및 전략을 다른 새로운 문제 상황에 전이될 수 있도록 해 준다.

이상의 연구를 토대로 알고리즘과 프로그래밍 지도 과정에서 문제해결력을 신장할 수 있는 문제해결 수업 모형을 [그림 1]과 같이 제시하였다. 제시한 프로그래밍을 이용한 문제해결 수업모형을 살펴보면, 문제의 이해, 계획의 수립, 계획의 실행, 그리고 반성의 4단계로 제시하였다. 좀 더 구체적으로 단계별로 설명하면 다음과 같다.

제1단계인 문제 이해에서는 문제를 해결하기 위해서 문제의 정확한 인식과 학습목표 탐색의 과정을 거친다. 첫째, 문제 인식 과정으로 문제를 잘 읽고 문제에 제시되어 있는 조건을 모두 고려하여야 하며 필요 없는 조건은 없는지, 필요한 정보가 빠져 있지는 않은지 등을 살펴본다. 문제는 다양한 해법을 가질 수 있고, 학생들이 도전할 만하고 재미있는 문제라고 인식시킨다. 둘째, 학습 목표 탐색과정으로 학습 목표를 찾고, 문제 해결을 위해서 사용하게 될 원리와 명령어를 탐색해 본다.

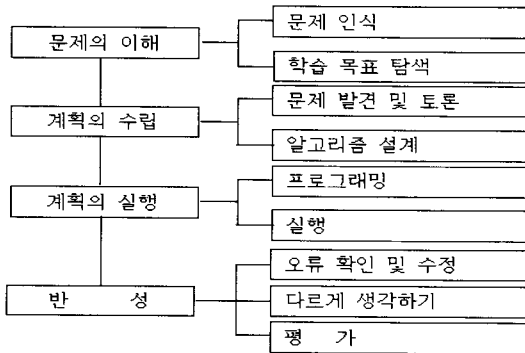


그림 1. 프로그래밍을 이용한 문제해결 수업모형

다음으로 계획 수립 단계에서는 문제의 이해 단계를 마친 후 이 단계에서 문제해결을 위한 아이디어를 탐색하기 위해 문제의 발견 및 토론과 알고리즘 설계를 실시한다. 첫째 문제 발견 및 토론과정으로 주어진 문제를 해결하기 위한 주요한 요소를 찾아내고, 문제해결을 위해서 다양한 도구를 활용하거나, 동료 간의 토론을 통해서 문제해결과정을 찾아내는 과정이다. 이 때 교사

는 학생들이 어려움을 겪을 때 적절한 발문을 제시하여 학생들이 스스로 어려움을 해결할 수 있도록 도와준다. 둘째, 알고리즘 설계 과정으로 문제 발견 및 토론 과정에서 찾아낸 문제해결 절차를 자연어나 의사코드, 순서도 중 하나를 이용하여 알고리즘을 설계하는 과정이다.

제3단계인 계획실행 단계에서는 수립한 계획을 실제로 실행하여 계획이 올바르게 되었는지 살펴보고 각 단계를 점검하는 단계로 프로그래밍과 실행의 과정을 거친다.

마지막 단계인 반성 단계는 문제해결력을 신장할 수 있는 중요한 부분으로 주어진 문제에 관하여 프로그램의 오류를 수정하고 결과를 확인하는 과정이다. 첫째, 오류 확인 및 수정과정으로 문제가 요구하는 답과 일치하는지를 확인하고, 오류가 발생하면 올바른 결과에 이를 때까지 반복적으로 수정을 한다. 이 과정에서 수렴적 사고와 문제해결력이 신장될 수 있다. 둘째, 다시 생각하기 과정으로 주어진 문제에 대해 다른 방법으로 그 결과를 유도할 수 있도록 함으로써 문제해결 전략에 대한 발산적 사고와 문제해결력이 향상된다. 셋째, 평가과정으로 문제해결과정에서 어려웠던 점과 서로 다른 해결방법을 찾은 학생들의 결과를 비교하여 보고, 문제해결 과정에 관한 토론을 통해서 스스로 평가하고 반성한다.

2. 학습 지도안

앞 절에서 기술한 프로그래밍 수업 모형을 기반으로 학습지도안 작성 사례를 제시한다. 본 논문에서 제시하는 사례는 조건문과 반복문을 이용하여 주어진 문제를 해결하는 과정과 작업 순서에 따라 결과가 어떻게 달라지는가에 관한 것을 가르치기 위한 학습지도안이다. 여기서 학습자는 프로그래밍 언어에서 변수 선언과 출력 명령어를 알고 있다는 가정 하에서 수업 모형을 적용하였다.

【문제】 혹시 방울꽃과 달팽이의 애절한 사랑 이야기를 들어보셨습니까? 방울꽃이 사랑하는 나비는 방울꽃의 달콤한 향기만을 좇아 날아들었고 방울꽃의 꽃잎이 지자 금방 날아가 버렸지요. 하지만 달팽이는 자신의 등껍질이 햇빛에 마르도록 방울꽃의 곁을 지켰다는

사랑 이야기입니다. 이 달팽이가 3m 깊이의 우물에 빠지게 되었습니다. 하루 이틀 꽃잎이 지는 방울꽃은 일주일이면 꽃잎이 모두 떨어져 시들어 버리게 됩니다. 달팽이가 낮 동안에 열심히 우물을 기어 올라가면 55cm를 올라가고 쉬는 밤중에는 13cm를 미끄러지게 됩니다. 과연 이 달팽이는 방울꽃이 시들기 전에 그 곁을 지킬 수 있을까요?

위의 문제를 통해 조건문과 반복문을 익히는 학습지도 과정을 기술하면 [표 7]과 같다.

표 7. 조건문과 반복문을 학습하기 위한 지도안

단계	주 제	교 육 내 용
문제 이해	문제 인식	달팽이가 우물을 기어 올라가는 과정을 생각해 보고, 달팽이가 원하는 시기에 방울꽃에 도착할 수 있을지에 대해 생각하고 여러 가지 방법을 동원하여 결과를 유추해 보도록 한다.
	학습목표탐색	주어진 문제를 해결하기 위해서 사용해야 할 명령문의 종류와 사용될 시점에 대해 탐색하고, 해결 과정이 무엇인지 말하도록 한다.
계획 수립	문제 발견 및 토론	주어진 문제에서 해결해야 할 문제와 해결 단계에 대해 발견 학습을 실시한 후 어려운 점이나 궁금한 사항에 관해서는 동료 간에 질문과 토론을 통해 해결해 가도록 한다. 교사는 학생들이 문제 발견 및 해결과정에서 방향할 때 질문으로 해결과정을 찾도록 유도한다.
	알고리즘설계	문제를 해결하는 알고리즘을 프로우차트 형식으로 작성한다. (그림 2) 참조
계획 실행	프로그래밍	설계된 알고리즘을 학습자에게 적합한 프로그래밍 언어로 바꾸는 과정이다. 이 때 문법적으로 어려운 부분은 참고서적을 참고할 수 있도록 한다. (그림 3) 참조
	실행	프로그래밍 언어로 작성한 프로그램을 직접 컴퓨터에서 실행해 본다.
반성	오류확인및수정	오류를 학습자 스스로 찾아 수정하고 재실행을 통하여 의도한 결과가 나올 때까지 반복적으로 실시하도록 한다.
	다르게생각하기	질문1) 조건문을 다른 것을 이용할 경우도 생각해 보도록 한다. 질문2) (그림 4)와 같이 알고리즘을 바꾼다면 결과는 어떻게 될 것인가 생각해 보도록 한다.
	평가	문제 해결 과정에서 어려웠던 점과 다른 알고리즘을 구현한 학생들과 문제해결 과정을 분석하고 토론하는 과정을 통해서 스스로 평가하고 반성한다.

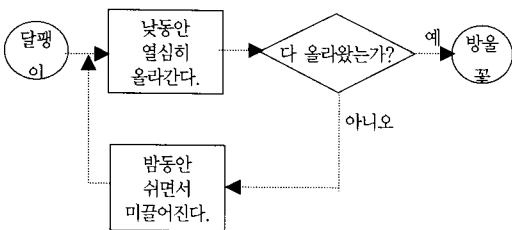


그림 2. 주어진 문제 해결을 위한 알고리즘

```

#include <stdio.h>
int main()
{
    int hi = 0, day = 0 ;
    hi = hi + 55;

    while( hi < 300) {
        hi = hi - 13;
        day = day + 1;
        hi = hi + 55; }
    printf("달팽이가 방울꽃 곁으로 가는데%d일 걸립니다.\n", day);
}
    
```

그림 3. 주어진 문제 해결을 위한 프로그램

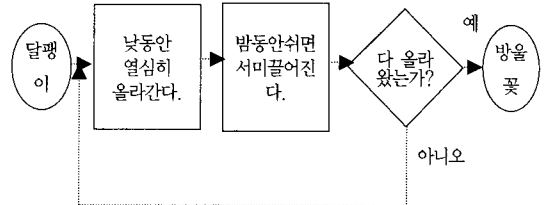


그림 4. 다른 의미로 수정한 알고리즘

V. 결론 및 토의

프로그래밍과 문제해결 능력간의 관계를 살펴보고, 프로그래밍 교육을 제한함에 있어 프로그래밍 교육 목표를 제시하고 이에 맞게 학습자의 인지 능력을 고려하며, 계열성을 가지도록 교육 내용을 제안하였다. 또한 프로그래밍을 통해 문제해결력을 신장할 수 있는 수업 모형을 개발하고, 그에 따른 학습지도안을 제시하였다. 먼저 프로그래밍과 문제해결력 간의 관계를 살펴본 결과 프로그래밍은 문제를 분석하여 프로그램을 작성하고 수정하는 과정에서 사고력 및 문제 해결력 신장에 크게 기여하고 있는 것으로 밝혀졌다. 컴퓨터교과의 프로그래밍 단원의 교육 목표와 내용은 선행 연구자들의 연구와 정보통신기술교육을 토대로 이루어졌다. 그리고 프로그래밍 교육의 세부적인 내용을 연속적이고 계열성을 가지도록 하기 위하여 본 연구에서는 컴퓨터 교육과정을 초등학교 3·4학년, 초등학교 5·6학년, 중학교, 고등학교 학교 급별로 조직하였다. 그리고 학교 급

별 교육내용을 제시함에 있어서 학습자의 인지 수준을 고려하고, 타 교과와의 연계 및 학습의 전이를 높이기 위하여 프로그래밍 단원과 학습의 전이가 가장 활발히 이루어지고 있는 수학교육과 연계하여 프로그래밍 관련 요소를 추출하여 연속적이고 계열성을 가지도록 제시하였다.

프로그래밍을 통한 문제해결 능력을 신장할 수 있는 수업 모형을 개발을 하기 위해 Polya의 문제해결 수업 모형과 문제해결 수업에서 효과를 검증된 수업모형을 토대로 문제 인식, 학습 목표 탐색, 문제 발견 및 토론, 알고리즘 구현, 프로그래밍, 실행, 오류 확인 및 수정, 다르게 생각하기, 평가 단계로 프로그래밍을 이용한 문제해결 수업 모형과 교수-학습지도안을 제시하였다. 본 연구의 결과는 다음과 같다. 첫째, 프로그래밍 교육은 학습자의 사고력 및 문제해결력 신장과 밀접한 관련이 있는 것으로 밝혀졌다. 둘째, 프로그래밍 언어 지도에 있어서 학습자의 실생활과 관련 있는 쉽고 흥미 있는 문제에서 출발하여, 단계적으로 문제해결 전략과 관련 있는 심화된 문제로 접근해야 한다. 그리고 타 교과와의 학습 전이가 이루어 질 수 있도록 지도 내용 선정에 관심을 가져야 한다. 셋째, 프로그래밍 언어 지도에서 기능적인 면에서 지도할 것이 아니라, 우선적으로 학습자의 인지 능력을 파악하고, 문제에 대한 인식과 알고리즘을 설계한 후 프로그래밍 작성 및 오류 수정 과정을 거치도록 한다. 이 때 문제해결과정에 대한 반성을 통해 문제를 다른 방법으로 해결할 수 있는가에 대한 문제의 재인식 및 새로운 해결과정의 토론을 통해 발산적 사고력과 문제해결력을 더욱 더 신장할 수 있을 것이다. 넷째, 문제해결력을 신장시킬 수 있는 수업 모형의 개발 및 적용에서 교사의 부단히 노력하는 자세와 문제에 대한 알고리즘적 사고, 프로그래밍 작성 능력을 요구한다.

향후 연구과제로는 본 연구에서 제안한 프로그래밍 교육 내용과 교수 모형을 각 학교 급별로 적용·실시하여 프로그래밍 교육 내용의 타당성을 구체적으로 검증하는 과정 등이 있으며, 또한 이러한 과정을 통해 학교 현장에서 실제 수업할 수 있는 보다 더 상세하고 구체적인 내용을 포함하는 교재를 연구 개발할 필요가 있다.

참고 문헌

- [1] 박미화, *목표모형에 기반한 중등 일반계 프로그래밍 교육과정 개발*, 한국교원대학교 대학원 석사논문, pp.1-2, 2006.
- [2] 김재철, *컴퓨터 교육의 현실과 대안*, 컴퓨터월드, p.50, 2005.
- [3] 김현준, *일반계 고등학교 컴퓨터과학 교육을 위한 정보·컴퓨터교과목의 성격 및 목표체계 연구*, 연세대학교 대학원 석사논문, 2005.
- [4] 박관우, *이성근, 웹기반 프로그래밍 교수/학습 시스템*, 대구대학원 석사논문, 2001.
- [5] 백영균, “컴퓨터 프로그래밍에 대한 심리학적 접근”, *교육공학연구*, Vol.4, No.1, pp.145-165, 1988.
- [6] 이정모, 이근효, “초보자의 C언어 학습과정에 대한 인지심리학적 분석 연구 : 프로그래밍 학습과정 동안의 은유 사용의 효과”, *인지과학*, Vol.9, No.4, pp.2-3, 1998.
- [7] 백영규, 우인상, “LOGO 프로그래밍의 수업방법이 문제해결력에 미치는 효과에 관한 연구”, *교육공학 연구*, Vol.9, No.1, pp.73-90, 1994.
- [8] 유인환, “창의적 문제해결력 신장을 위한 로봇 프로그래밍의 가능성 탐색”, *교육과학연구*, Vol.36, No.2, pp.109-128, 2005.
- [9] 박성현, *컴퓨터과학을 위한 수학교육*, 중앙대학교 대학원 석사논문, 2002.
- [10] 정치봉, *중등 수학 및 컴퓨터과학 교육에서 컴퓨터 교육의 연결성 및 통합성*, 한국수학교육학회지 시리즈E 수학교육 논문집, Vol.10, pp.303-324, 2000.
- [11] 이종영, *컴퓨터 환경에서의 수학 학습-지도에 관한 교수학적 분석*, 서울대학교 대학원 석사논문, 1999.
- [12] 문선영, *한국과 인도의 컴퓨터 교육과정 비교를 통한 중학교 컴퓨터 교육과정 모형 개발*, 신라대학교 대학원 석사논문, 2005.
- [13] 윤기창, *일반계 고등학교 컴퓨터교육과정 개선에 관한 연구-프로그래밍 중심의 컴퓨터과학 교*

- 육과정-, 연세대학교 대학원 석사논문, 2004.
- [14] 교육인적자원부, 초·중등학교 정보통신기술 교육 운영지침 해설서(안), 선명인쇄주식회사, pp.30-31, 2006.
- [15] 신은미, 김현철, 일반계 고등학교에서의 컴퓨터 교과 교육과정에 대한 현황과 개선방향, 정보처리학회 학회지, Vol.9, No.5, pp.26-34, 2002.
- [16] 우정호, *어떻게 문제를 풀 것인가*, 서울: 천재교육, 1986.
- [17] 박미숙, *Logo을 이용한 문제해결력 신장을 위한 수업모형설계*, 동국대학교 대학원 석사논문, pp.19-21, 1991.
- [18] J. G. Brookshear, 황중선, 광덕훈, 이상근, *컴퓨터과학총론 7판*, 홍릉과학출판사, 2004.
- [19] 교육인적자원부, *초등학교 수학 교사용지도서*, 대한교과서주식회사, 2002.
- [20] 교육부, *수학과 교육 과정(교육부 고시 제 1997-15 호 별책 8)*, 대한 교과서 주식회사, 1997.

유 관 희(Kwan-Hee Yoo)

정회원



- 1985년 8월 : 전북대학교 전산 통계학과(공학사)
 - 1988년 2월 : 한국과학기술원 전산학과(이학석사)
 - 1995년 8월 : 한국과학기술원 전산학과(공학박사)
 - 1988년 ~ 1997년 : (주)데이콤 종합연구소 선임연구원
 - 2003년 ~ 2005년 : 카네기멜론대학교 로보틱스 연구소 방문교수
 - 1997년 ~ 현재 : 충북대학교 컴퓨터교육과, 정보산업공학과 교수
- <관심분야> : 컴퓨터 그래픽스, 덴탈 및 메디칼 그래픽 처리, 로보틱스, 교육용 콘텐츠

저 자 소 개

박 영 미(Young-Mi Park)

정회원



- 1991년 : 계명대학교 전자계산학과(공학박사)
- 2007년 : 충북대학교 정보·컴퓨터교육대학원(교육학 석사)
- 1991년 ~ 현재 : 충북옥천여자중학교

<관심분야> : 컴퓨터교육, 정보통신윤리