# 대학 프로그래밍 강좌를 위한 프로그래밍 교육 프레임워크

최혀종\*

#### 요 얃

본 연구는 대학에서 교육되고 있는 프로그래밍 강좌를 위한 프로그래밍 교육 프레임워크를 제안 한다. 먼저 프로그래밍에 관한 국내외의 다양한 심리학 및 교육학 관련 논문을 살펴본 결과, 몇 개 의 논문에서 초보 프로그래머와 전문 프로그래머 사이에 존재하는 정신 모델, 프로그래밍 지식과 전략에 관한 차이를 확인할 수 있었다. 또한 초보 프로그래머가 겪는 많은 문제점들도 확인할 수 있었다. 그래서 이를 우선 정리하여 프로그래밍 교사에게 실제적인 도움이 될 수 있는 시사점을 도 출하였고, 초보 프로그래머의 실력을 향상시키기 위해 프로그래밍 개발 과정, 프로그래밍 학습 요 소, 교수학습 방법을 세 개의 차원으로 하는 프로그래밍 교육 프레임워크를 제안한다. 이 프레임워 크는 프로그래밍 강좌의 개발과 운영, 평가를 위한 가이드라인이 될 것이다.

주제어: 프로그래밍 교육 프레임워크, 정신 모델, 프로그래밍 지식과 전략

## The Programming Education Framework for Programming Course in University

Hyunjong, Choe<sup>†</sup>

#### **ABSTRACT**

In this paper I present the framework for programming education relating to the programming course in university. First of all, I identify the literature of the psychological and educational study of programming in domestic and foreign country. Some outstanding papers in foreign country show that the mental model, programming knowledge and strategies are the important difference between novice and expert programmers. And various problems experienced by novice programmers are identified. I summarise this material and suggest some practical implications for programming teachers in their teaching activity. The framework for programming education presented by this work has three dimensions of program development process, programming learning factors and teaching and learning methods for programming to develope the novice's programming skill. It helps the programming teacher to design, implement and evaluate their programming course with the guideline of programming activities.

Keywords: Programming Education Framework, mental model, programming knowledge and strategies

#### 1. 서 론

컴퓨터 과학 교육의 가장 중요하고, 관심있는 분야의 하나는 바로 프로그래밍 교육이다. 사람들 이 언어로 자신의 생각을 표현하고 다른 사람들 과 소통하듯이 우리는 프로그래밍 언어를 통해 컴퓨팅의 세계와 소통할 수 있기에 프로그래밍은 컴퓨터 과학 및 공학에서 중요한 학문의 분야이 다[1]. 그래서 프로그래밍 과목은 중등학교의 정 보·컴퓨터 교사 양성을 위한 컴퓨터 교육과의 전 공 필수 과목 중의 하나이고, 교원자격증 무시험 검정을 위한 기본 이수 과목이다. 국내의 사범대 학 컴퓨터 교육과는 대부분 <표 1>에서 제시한 바와 같이 프로그래밍 강좌를 1학년부터 시작하 고 있다. 프로그래밍 1과 프로그래밍 2의 형태로 두 학기에 개설하는 학과도 있고, 프로그래밍 강 좌의 연계 강좌로 자료 구조나 알고리즘을 구성 하는 학과도 많다.

<표 1> 컴퓨터교육과의 프로그래밍 개설 현황

학교	개설 과목	개설학기
A대학	프로그래밍언어및실습1	1학년 1학기
C대학	기초컴퓨터프로그래밍	1학년 1학기
K대학	프로그래밍1	1학년 1학기
K대학	프로그래밍 1	1학년
M대학	컴퓨터프로그래밍	1학년 1학기
S대학	전산학개론및프로그래밍1	1학년 1학기
S대학	기본프로그래밍	1학년

하지만 프로그래밍은 가르치고 배우는 모든 사람들이 가장 어려워하고, 가장 싫어하는 컴퓨터 관련 과목의 하나이기도 하다. 실제로 프로그래밍 강좌의 학습 탈락율은 심각한 편이다. 외국 연구의 경우 Dehnadi & Bornat는 영국의 컴퓨터 과학 학부 대학생 중 30-60%가 프로그래밍 강좌에서 탈락하고 있는 것으로 보고하였다[2]. 우리나라역시, 프로그래밍 강의를 해 본 교수라면 외국의탈락 사례와 비슷한 수치의 탈락자가 생기는 것을 경험할 수 있을 것이다. 즉 프로그래밍 과목은컴퓨터 과학 및 공학 교육에서 가장 핵심 과목임에도 불구하고, 학생들은 가장 어려운 과목 중의하나로 인식하여 학습 탈락율이 높은 과목중의

하나이다.

이에 프로그래밍 교육에 관한 다양한 국내의 연구들이 현재 진행되고 있다. 본 논문의 2장에서 는 국내외에 연구되고 있는 다양한 프로그래밍 관련 연구를 정리하여 프로그래밍 교육에 참고할 수 있는 여러 교수 전략을 요약하여 제시하였다. 하지만, 대부분의 연구들이 프로그래밍 교육의 미 세 전략에 대해서는 그동안의 연구 결과로 잘 정 리되고 있지만 실제 프로그래밍 강좌를 설계할 때 참조할 수 있는 거시 전략에 대해서는 연구가 미흡한 상황이다. 이에 본 연구는 프로그래밍 교 육에 필요한 요소들을 추출하여 이를 프로그래밍 교육 프레임워크로 제시하였다. 3장에서는 프로그 래밍 교육 전문가와 컴퓨터 교육 전문가로 구성 된 전문가 집단의 설문을 통해 프레임워크 요소 를 추출하고, 이를 통해 프로그래밍 교육 프레임 워크를 제안하였다.

#### 2. 관련 연구

#### 1.1 국내 프로그래밍 교육 관련 연구

국내의 프로그래밍 교육 관련 연구는 크게 수업 모형, 프로그래밍 교육 내용, EPL(Educational Programming Language) 및 로봇을 이용한 프로그래밍 교육 등으로 구분할 수 있다.

프로그래밍 수업 모형 개발 연구에서 박영미, 유관희는 프로그래밍을 이용한 문제해결 수업 모형을 제안하였다[3]. 특히, 이 연구는 초중등학교의 수학 교과의 내용과 프로그래밍 학습 요소를서로 관련지어 학교급별 프로그래밍 교육 내용을제시하였다. 김갑수는 초등학교에서 교육되는 프로그래밍 교육이 단순한 행동주의적 교육 심리에의한 방법이 적용되어, 본연의 창의성과 논리성향상에 도움이 되지 못하고 있다고 지적했다[4].이에 소프트웨어 생명주기를 기반으로 문제 이해, 설계, 구현, 실험, 유지보수 단계에 따라 교육되는 방법을 제안하였다. 배학진 외는 PBL을 기반으로한 스크래치 프로그래밍 교수학습 모형을 제안하였다[5]. 이 연구에 의하면 문제 인식, 문제 해결계획 세우기, 탐색하기, 해결책 발견하기, 발표 및

평가의 단계로 수업 모형을 설계하여 실제 초등 학교 학생들에게 적용하여 유의미한 논리적 사고 력 및 문제 해결력 향상을 가져왔다. 또한 김병욱 외는 프로그래밍 수업에서 PBL이 효과적인 교수 학습 모형임을 밝혔는데, 특히 학습 동기가 전반 적으로 향상된 것을 밝혔다[6]. 또한 PBL 기반의 프로그래밍 수업 모형에서 학습 양식의 차이에 따른 학습 동기의 차이를 있음을 밝혀냈다.

김미량은 프로그래밍 교육에 적용 가능한 다양 한 교수학습 방법을 제안했는데, 그 대표적인 교 수방법으로 강의법/설명법, 시연, 실습, 질문법과 토론법을 제시하였다[7]. 또한 효과적인 방법으로 프로젝트, PBL, 협동학습, 동료 교수, 사례 연구 등을 제안하였다. 또한 수업 모형으로 문제 규명, 문제의 연구 및 분석, 다양한 가능성 검토, 실천, 반성 과정을 제시하였다. 또한 교수자의 역할을 촉진형 교수자로 해야 된다고 했다. 김미량의 연 구는 컴퓨터 프로그래밍 학습에 적용 가능한 다 양한 교수학습 방법을 제안했다는 것에 그 의의 가 있다.

프로그래밍 교육 내용을 제시한 연구로 문외식 은 초등학생의 논리적 사고력 및 문제 해결 능력 향상을 위해 컴퓨터 프로그래밍 교육이 필요하며. 이를 위해 비주얼베이직 언어의 교육과정 모델을 제시하였다[8]. 제시된 내용은 비주얼 베이직의 교 육 내용을 60차시 정도로 구성하여 제시하였으나, 교육 내용만을 제시하였다. 유관희는 초중등학교 의 수학 교과의 내용을 기반으로 중등학교의 컴 퓨터 프로그래밍 내용을 제시하였다[9].

EPL 및 로봇을 적용한 프로그래밍 교육에 관한 연구로 최해심 외는 EPL인 두리틀에 친숙한 명령 어로 프로그래밍을 하도록 환경을 제공해 준 결 과, 표현 방법의 차이에서 발생했던 학생들의 오 류를 크게 줄일 수 있었다고 발표하였다[10]. 유인 환, 채재호는 초등학교 프로그래밍 교육에 로봇을 활용한 교육 방안을 제시하였는데, 로봇을 활용한 프로그래밍 교육이 효과적임을 제시하면서 이를 위해 로봇보다 프로그래밍 교육에 초점을 둔 프 로그램 개발이 필요하며 불필요하거나 어려운 내

용은 개선하거나 줄여야 한다고 제안하였다[11]. 조성환 외는 프로그래밍 교육에서 학습자가 어려 워하는 프로그래밍 지식(문법)이 적게 필요한 EPL인 스크래치 언어를 적용했을 때의 프로그래 밍 태도에 미치는 효과를 연구하였다[12]. 그 결과 스크래치를 적용한 CPS(Creative Solving) 교육이 프로그래밍 태도와 문제 해결력 에 긍정적 영향을 미친 것으로 발견되었다. 김수 환 외는 2007 개정 교육과정의 정보 과목에 제시 된 프로그래밍 교육을 위해 필요한 EPL을 분석하 여 실제 학교에서 적용 가능한 방안을 제시하였 다[13]. 26개의 EPL을 교육과정에 제시된 학습 요 소와 교차 분석하여 실제 적용 가능한 EPL을 제 시하는데 그 의의가 있다.

또한 프로그래밍 교육에 대한 인식 및 적용 전 략에 관한 연구도 있는데, 김종훈은 초등학교 학 생들의 창의성 교육을 위해 프로그래밍 교육이 필요하며, 이를 위해 교육 시간 확보, 간단한 프 로그래밍 언어 규칙 정리, 교재 개발과 교사의 양 성 등 개략적인 전략들을 진술해 놓았다[14].

차승은 외는 예비교사를 대상으로 실제 프로그 래밍 교육을 실시해 보고, 이에 대한 인식 변화를 살펴보는 연구로 수업에 참여한 학생들은 프로그 래밍의 개념과 교육의 필요성에 대한 인식이 변 화되었음을 제시하였다[15].

최근 프로그래밍 교육 관련 국내의 연구는 교 수학습 방법의 개발과 EPL 및 로봇을 적용한 연 구가 다수를 차지한다. 하지만, 교수학습 방법의 연구는 문제해결수업이나 프로젝트 모형을 기본 으로 변형된 형태의 교수학습 방법을 개발하여 현장에 적용하는 연구가 많다. 또한 EPL 및 로봇 을 프로그래밍 교육에 적용했을 때의 효과성에 관한 연구도 많은데, 특수한 교육 상황을 전제로 했기 때문에 일반적인 프로그래밍 교육에 참조할 수 있는 전략과 교수학습 방법 연구가 필요하다.

#### 1.2 국외 프로그래밍 교육 관련 연구

국외의 프로그래밍 교육 관련 연구는 국내와

달리 초보 프로그래머의 사례와 특성을 연구하고 그에 적절한 교수학습 전략을 제안하는 연구가 돋보인다.

우선 초보 프로그래머와 전문 프로그래머의 특성을 비교하여 초보 프로그래머의 특성을 제시한 연구가 있다. 초보 프로그래머는 전문 프로그래머 가 가지고 있는 지식과 기술이 부족한 상태의 프로그래머라고 할 수 있다[16]. 즉 처음 프로그래밍을 배우는 학습자로 프로그래밍 언어에 대한 지식과 기술을 배우는데, 초보 프로그래머가 가지고 있는 일반적인 특성을 전문 프로그래머의 지식과 기술에 비교하여 정리하면 다음과 같다 [17][18][19][20].

- 프로그래밍 언어의 중요 개념(변수, 반복, 배열 등)을 정확히 이해하지 못한다.
- 프로그램의 구조를 파악하기보다 코드(code) 한 줄을 해석하는데 집착한다.
- 프로그램을 계획하고 설계하는데 아주 적은 시간을 소비한다.
- 프로그램 전체를 재구조화하기 보다 지협적인 버그(bug)를 수정하는데 집착한다.
- 프로그램을 테스트하고 코드를 트래킹 (tracking)하는데 미흡하다.
- 배운 프로그래밍 지식을 프로그램에 정확하게 적용하지 못한다.
- 일상 언어와 프로그래밍 언어의 차이를 구별 하지 못한다.
- 프로그램을 이해하는 정신 모델(mental model)이 부족하다.
- 초보 프로그래머의 프로그래밍 지식은 아주 약하다(fragile).

정신 모델은 프로그래머가 정신 속에 생각하고 있는 프로그램의 현재 상태를 지칭한다. 예를 들어 프로그램을 개발하다 보면 의도하지 않았던 방식으로 실행되는 프로그램을 만나게 된다. 이때 프로그래머는 코드가 실제 실행하는 방식대로 변수와 제어의 흐름을 머리 속에 생각하는데, 이를 정신 모델이라고 한다. 정신 모델은 학습자가 프로그램을 이해하는데 매우 중요한 요소로 프로그

래밍 학습의 인지적 과정(cognitive process)을 분석하는데 유용하게 사용된다[21]. Perkins과 Martin은 초보 프로그래머의 프로그래밍 지식을 깨지기 쉬운 약한(fragile) 지식이라고 표현했다[17]. 이는 습득한 지식이 쉽게 깨어져 쓸모없게된다는 의미로 사용한 것으로 초보 프로그래머들이 프로그래밍 지식을 쉽게 잊고, 배웠지만 적용하지 못하고, 잘못 사용되는 세 가지 경우를 표현한 것이다.

또한 Du Boulay, Spohrer와 Soloway, Détienne 는 특정 프로그래밍 언어를 학습할 때 초보 프로그래머가 공통적으로 실수하는 몇 가지 일반적인 경우를 다음과 같이 제시하고 있다[22][23][24].

- 변수를 업데이트하는 것보다 초기화하는 것을 더 어려워한다.
- 반복과 조건문에서 자주 실수를 한다.
- 반복문에서 조건 변수가 변하는 것과 같이 코
  드 안에 숨어있는 상황을 특히 어려워한다.
- 재귀의 이해와 적용은 많은 학생들이 어려워 한다.
- 반복문을 먼저 학습한 뒤 재귀 함수를 배우면 재귀를 쉽게 이해하는데, 그 반대는 그렇지 않 다.
- "123"과 123와 같이 서로 비슷해 보이지만 전혀 다른 의미를 지닌 것을 어려워 한다.
- 포인터, 동적 메모리 할당, 참조, 다형성, 다중 정의 등 의 개념은 교사와 학생 모두에게 어렵다.
- 프로그래밍 언어의 구문이나 개념적 이해보다 더 어려워하는 것이 바로 프로그램의 설계이 다.
- 종이와 연필로는 문제를 해결할 수 있지만, 컴 퓨터 프로그램으로 해결하는 것은 어려워한다.

두 번째로 프로그래밍 지식과 전략에 관한 연구에서는 Davies는 프로그래밍 교육에 프로그래밍 지식과 프로그래밍 전략을 서로 구분하여 함께 가르쳐야 한다고 제안하였다[19]. 프로그래밍지식이란 프로그래밍 언어의 구문과 의미 등 프로그래밍에 관한 지식이고, 프로그래밍 전략이란

프로그래밍 지식을 사용하는 방법으로 프로그램 을 작성하는 기술이라고 할 수 있다.

일반적으로 프로그래밍을 잘하기 위해서는 프 로그래밍 지식이 먼저 선행되어야 하기 때문에 대부분의 프로그래밍 입문 과정이나 초보 프로그 래머를 교육하는 과정은 특정 프로그래밍 언어에 대한 학습 내용으로 구성되어 있다. C와 Java와 같이 최근 많이 사용되는 프로그래밍 언어에 대 한 학습 교재만 보더라도 C와 Java의 언어적 특 성을 배우는 '지식' 기반의 학습 내용으로 구성되 어 있으며, 실제 대학이나 사범대학의 컴퓨터 관 련 학과의 프로그래밍 강의 역시 프로그래밍 언 어에 대한 지식을 가르치고 있는 실정이다.

하지만 프로그래밍 지식에 한정된 교육은 지식 자체만을 가르치게 되고, 프로그래밍 지식이 실제 프로그램 개발에 어떻게 사용되고 적용되는지에 대한 고민은 하지 않게 된다. 따라서 초보 프로그 래머들은 변수, 입출력, 제어문이 무엇인지는 알 고 있지만, 이를 실제 프로그램과 문제 상황에 적 용하기 못하고 있기 때문에 프로그래밍이 어렵고 재미없게 되는 것이다.

프로그래밍 전략은 결국 프로그램의 이해 정도 에도 큰 역할을 하는데, Widoski와 Eyferth는 초 보 프로그래머와 전문 프로그래머의 프로그래밍 전략을 비교한 다음, 이 두 집단이 제시된 프로그 램을 얼마나 빠르고 정확하게 이해하는지 그 차 이를 비교해 보았다[25]. 뛰어난 프로그래밍 전략 능력을 가진 전문 프로그래머 집단이 구조화된 프로그램과 구조화되지 않은 프로그램 모두를 빠 르고 정확하게 이해한 반면, 초보 프로그래머는 그렇지 못했다.

세 번째로 프로그램의 이해와 작성에 관한 연 구에서는 프로그램 이해와 프로그램 작성의 차이 를 구별해야 한다고 한다. 프로그램의 이해란 주 어진 프로그램을 살펴본 후 어떻게 동작하는지를 아는 것이고, 프로그램의 작성이란 특정 작업이나 문제를 해결하기 위해 프로그램의 전체나 일부분 을 실제 작성하는 것을 의미한다. Brooks는 프로

그램의 이해에 관한 모델을 처음 제안하였다[26]. 프로그래밍은 문제 영역을 프로그래밍 영역으로 변환하는 과정으로, 프로그램의 이해를 위해 하양 식(top-down), 가설 기반(hypothesis-driven)의 과정이 필요하다고 하였다. 즉 프로그램을 코드의 줄 단위로 이해하는 것이 아니라, 프로그램 전체 를 특정 문제 해결을 위한 가설을 기반으로 이해 해야 한다고 했다. 프로그램의 이해에 관한 연구 는 Brooks 이외에도 Davies, Mayrhauser와 Vans, Shneiderman과 Mayer 등의 연구에서도 언 급하고 있다[19][27][28].

프로그램의 개발에 관한 연구는 이해에 관한 연구보다 적은 편인데, 대표적으로 Rist가 프로그 램의 작성에 관한 모델을 제안하였다[29]. 정보 처 리 이론에서와 같이 사람의 지식은 노느(node)와 링크(link)로 구성되어 저장되는데, 프로그램을 작 성할 때 우선 비슷한 설계 전략을 지식에서 검색 하여 프로그램을 설계하게 된다. 초보 프로그래머 의 경우 원하는 프로그램 설계 전략과 비슷한 지 식이 없으면 새롭게 개발해야 되기 때문에 전문 프로그래머보다 어렵게 느껴지게 된다.

네 번째로 프로그래밍 학습 요소에 관한 연구 에서 Du Boulay는 학생들이 종이에 기술된 프로 그램과 실제 실행되는 시스템의 차이를 이해하기 위해서는 많은 시간이 필요하다고 이야기하고 있 다[22]. 왜냐하면 초보 프로그래머의 경우, 바로 앞 명령에 의해 실행된 그 상태를 기반으로 실행 되는 다음 명령어의 행위를 쉽게 이해하기 어렵 기 때문이다. 따라서 프로그래밍 언어를 간략화하 고, 그 실행 과정을 쉽게 눈으로 확인할 수 있는 개념적 기계(notional machine)가 필요하다고 주 장하였다. 1980년대에 입문용 프로그래밍 언어로 사용된 LOGO, 최근 교육용 프로그래밍 언어로 많이 사용되고 있는 Scratch가 대표적인 예라고 할 수 있다. LOGO는 거북이가 명령어를 통해 움 직이고, 프로그램의 현재 상태와 변화하는 과정을 시각적으로 직관적으로 보여주고 있고 Scratch 역 시 블록으로 구성된 명령어와 직관적인 인터페이 스로 인해 개념적 기계의 전형적인 예라고 할 수 있다.

교육용 프로그래밍 언어, 또는 입문용 프로그래밍 언어의 요구 조건으로 언급하고 있는 요소들은 다음과 같다[30][31].

- 읽기 쉬운 문법(readable syntax) : 주요 문법 적 키워드들이 쉬운 기호로 표현
- 고수준 언어(high level) : 프로그래밍 입문기의 학생들에게 어려운 개념들(포인터 등)은 제외
- 작은 규모(small size): 짧은 시간에 쉽게 배워 바로 활용할 수 있는 정도로 적은 문법과 의미로 구성
- 핵심 개념(core concept): 핵심 개념이 정확하고 일관성 있게 제시되고, 제시된 핵심 개념만 으로도 실제 프로그래밍이 가능
- 연관성(connection) : 배운 언어와 실제 산업 현장에서 사용되는 언어가 서로 연관

하지만 많은 대학의 프로그래밍 강좌에서 교육학적 이유에서 보면 적절한 교육용 프로그래밍 언어를 선택하지 않고, 실제 산업 현장에서 사용되고 있는 어려운 프로그래밍 언어를 선택하여가르치고 있다. 실제 산업 현장에서 사용되고 있다는 이유와 C와 Java가 어렵고 복잡하기 때문에일찍부터 이 언어를 배우기 시작해야하기 때문에입문용 프로그래밍 언어로 선택하고 있다고 한다.하지만, 학부에서 일찍부터 배운 프로그래밍 언어가 그 학생이 졸업한 이후에도 계속 현장에서 사

용되는 예도 있지만, 그렇지 않은 예도 있다는 것을 간과해서는 안된다.

국외의 연구는 국내의 연구와 달리 오랜 사례기반의 연구를 통해 실제 프로그래밍 교육에 적용할 수 있는 수업 전략을 제시하고 있으며, 특히 프로그래머의 심리적, 교육적 특성을 분석하여 프로그래머의 프로그래밍 지식과 전략, 정신 모델을모두 한꺼번에 교육해야 효과적이라고 이야기하고 있다. 또한 프로그래밍 환경으로 최근 국내에서도 많이 연구되고 있는 EPL도 포함되는 개념적기계의 도입을 필연적으로 포함해야 한다고 한다.

#### 3. 프로그래밍 교육 프레임워크

#### 1.3 프로그래밍 교육 전문가 설문

Robins은 프로그래머가 갖추어야 할 세 가지 요소로 프로그래밍 지식과 전략, 정신 모델 (mental model)을 제시하고, 프로그램의 개발 과정인 설계, 개발, 평가의 단계에서 프로그래머에게 필요한 학습 요소들을 정리하여 <그림 1>과같이 프로그래밍 프레임워크를 제시하였다[32].이 프로그래밍 프레임 워크는 프로그래밍과 관련된 다양한 문제들을 정리하여 프로그래머의 자질과 프로그래밍 과정의 틀에서 제시한 것에 그 의의가 있다. 예를 들어 프로그램을 설계하는 단계에서 프로그래머는 설계 방법, 알고리즘 설계, 전형적인 설계 방법들에 대한 지식이 필요하다.

	Knowledge	Strategies	<u>Models</u>
<u>Design</u>	of planning methods,	for planning,	of problem domain,
	algorithms design,	problem solving,	notional machine
	formal methods	designing algorithms	
Generation	of language, libraries, environment tools	for implementing algorithms, coding, accessing knowledge	of desired program
Evaluation	of debugging tools and methods	for testing, debugging, tracking/tracing, repair	of actual program

본 연구에서는 프로그래밍 교육에 관한 프레임워크를 개발하고자 Robins의 프로그래밍 프레임워크에 제시된 요소들을 프로그래밍 강좌의 학습요소로 제시할 수 있는지의 여부에 대해 전문가설문을 실시하였다. 더불어 프로그래밍 교육에 관한 프레임워크 설계를 위해 교수학습 방법에 대한 설문도 함께 실시하였다. 전문가 설문은 사범대학 컴퓨터교육과에 재직하고 있는 프로그래밍전공 교수 4명과 컴퓨터 교육 전공자 4명을 대상으로 메일을 통해 실시하였다. 프로그래밍학습요소 추출을 위한 설문은 프로그래밍 전공 교수와 컴퓨터 교육 전공자 모두에게 설문을 실시하였고, 교수학습 방법에 대한 설문은 컴퓨터 교육전공자만을 대상으로 실시하였다.

우선 프로그래밍 교육에 Robins가 제시한 프로그래밍 지식, 전략, 정신 모델이 필요하느냐에 대한 설문에 전문가 8명의 설문 결과는 <표 2>와같다. 설문에 참여한 모든 전문가가 프로그래밍교육에 지식, 전략, 모델 세 가지가 필요하다고응답하였는데 특히 프로그래밍 전략에 대한 필요성을 더 강하게 선택했다.

<표 2> 프로그래밍 교육의 학습 요소 설문 결과

프로그래밍 학습 요소	필요 하지 않다.	조금필 요하지 않다.	보통 이다.	조금 필요 하다.	매우 필요 하다.
프로그래밍 지식		1	1	4	2
프로그래밍 전략				2	6
정신 모델		1		6	1
합 계		2	1	12	9

다음으로 프로그래밍 지식과 전략에 대한 설문을 실시하였다. 정신 모델은 학습 요소를 추출하기 어려운 정신 작용이기 때문에 설문에서는 제외하였다. 프로그래밍 교육에 필요한 프로그래밍지식에 해당하는 학습 요소를 선별하는 설문에전문가 8명의 설문 결과는 <표 3>과 같다.

<표 3> 프로그래밍 지식의 학습 요소 설문 결과

학습 요소	필요 하지 않다.	조금 필요 하지 않다.	보통 이다.	조금 필요 하다.	매우 필요 하다.
데이터 형			2	4	3
입출력 및 배정(assignme nt)			2	5	1
제어 구조				3	5
함수/라이브러 리				2	6
특징적 개념 : 포인터, 상속, 메모리 관리 등		1	1	3	3
개발 도구			4	3	1
디버깅 방법				7	1
프로그램 설계 방법 : UML 등		1	1	4	2
알고리즘			1	1	6
합 계		2	11	32	28

설문에서 제시된 학습 요소 이외에 배열, 모듈, 동적 할당, 파일 처리 등도 포함되어야 한다는 의 견이 있었다.

세 번째 설문으로 프로그래밍 교육에 필요한 프로그래밍 전략에 해당하는 학습 요소를 선별하 는 설문에 전문가 8명의 설문 결과는 <표 4>와 같다.

<표 4> 프로그래밍 전략의 학습 요소 설문 결과

학습 요소	필요 하지 않다.	조금 필요 하지 않다.	보통 이다.	조금 필요 하다.	매우 필요 하다.
문제 해결				3	5
알고리즘 디자인			1	1	6
알고리즘의 적용			1	2	5
코딩			4	2	2
프로그램 테스트 및 디버깅			1	6	1
코드 트래킹/유지보수			3	4	1
합 계			10	18	20

설문에서 제시된 학습 요소 이외에 문제 분석 방법, 알고리즘 분석, 등도 포함되어야 한다는 의 견이 있었다. 네 번째 설문은 프로그래밍 지식과 전략에 적절한 교수학습 방법에 대한 설문이다. 이 설문은 컴퓨터교육 전공자 4명을 대상으로 실시한 것으로 중복 응답이 가능하도록 했으며, 그 결과는 <표 5>와 같다.

제시된 교수학습 방법 이외에 알고리즘을 추적 하는 과정에서 역할 놀이를 적용할 수 있다는 의 견도 추가로 제시되었다.

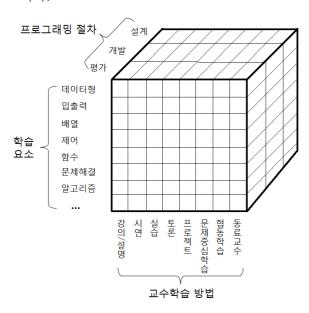
#### 1.4 프로그래밍 교육 프레임워크

프로그래밍 교육을 위한 학습 요소와 그 학습 요소에 적절하다고 판단되는 교수학습 방법을 전문가 설문을 통해 추출하였다. 전문가의 검토를통해 제시된 프로그래밍 지식 영역과 프로그래밍전략 영역의 학습 요소를 교수학습 방법과 서로교차하여 참조할 수 있는 프로그래밍교육 프레임워크를 제시하면 <그림 2>와 같다. 프로그래밍은 결국 설계, 개발, 평가라는 세 단계의 개발 절

<표 5> 프로그래밍 교수학습 방법의 설문 결과

학습 요소		강의/ 설명	시연	실습	토론	프로 젝트	PBL	협동 학습	동료 교수	사례연 구
	데이터 형	2	1		1		1			
	입출력 및 배정(assignment)	2	1	1			1			
	제어 구조		1	3			1		1	
	함수/라이브러리	1	1	2		1	1			
프로그래밍 지식	특징적 개념 : 포인터, 상속, 메모리 관리 등	3	1	2			1			1
	개발 도구	1	2	2						
	디버깅 방법	1		2	1		2	2	1	
	프로그램 설계 방법 : UML 등	1		1		1	2	1		1
	알고리즘			2	1	1	2	1		
	문제 해결	1		2	1		2	1		1
	알고리즘 디자인	1		2			2	1		
	알고리즘의 적용			2		2		1		
프로그래밍 전략	코딩	2	1	4						1
	프로그램 테스트 및 디버킹			2		1	2	2	1	
	코드 트래킹/유지보수	1		3		1	1		1	
합계		16	8	30	4	7	18	9	4	4

차에 따라 이루어지고, 이를 위해 프로그래머는 프로그래밍 지식과 전략, 정신 모델이 필요하다. 프로그래밍 개발 절차에 따라 필요한 학습 요소가 생성되고, 이 학습 요소를 교수학습하는데 필요한 방법을 서로 교차 참조할 수 있는 프레임워크이다.



<그림 2> 프로그래밍 교육 프레임워크

예를 들어 프로그래밍 절차의 평가 영역에 필요한 디버깅 학습 요소를 학습할 때, 실습, PBL, 협동 학습 등을 적용할 수 있을 것이다. 교수학습방법의 선택은 <표 5>를 기준으로 선택할 수 있다.

본 연구의 프로그래밍 교육 프레임워크를 기본으로 전문가 설문에서 조사된 학습 요소와 교수학습 방법 중에서 가장 많이 선택된 요소들을 추출하여 프로그래밍 교육의 15주 강의 내용을 추출하면 <표 6>과 같다.

<표 6> 프로그래밍 강의의 예시

주	학습 영역	학습 요소	절차	방법
1		강의 소개	ı	-
2	전략	문제 해결 방법/절차	설계	PBL
3	지식	데이터형	개발	강의/실습
4	지식	입출력	개발	강의/실습
5	지식	제어 구조	개발	실습
6	지식	배열	개발	강의/실습
7	지식	함수/라이브러리	개발	실습
8	전략	알고리즘 디자인	설계	PBL
9	전략	알고리즘 적용	설계	프로젝트
10	지식	프로그램설계방법	설계	PBL
11	지식	개발도구/코딩	개발	시연/실습
12	전략	테스트/디버깅	평가	실습/협동
13	전략	코드 트래킹/유지보수	평가	실습
14	전략	프로젝트		프로젝트
15		기말시험	-	_

#### 4. 결 론

프로그래밍 지식 교육에 관련된 것은 이미 많은 교재, 강의, 인터넷의 자료 등이 잘 구조화되어 제공되고 있다. 하지만 이 지식을 이용하여 어떻게 프로그램에 적용하고 개발할 것이냐를 결정하는 프로그래밍 전략은 프로그래밍 강좌에서 가장 중요시하게 생각되고 있지만 이제까지 프로그래밍 지식보다 관심을 받지 못했다. 이제는 프로그래밍 전략이 무엇이고, 이를 위해 필요한 정신적, 기술적 요소로 무엇이 필요하고 어떻게 교육해야 되는지 고민할 시기가 된 것이다.

본 논문에서는 프로그래밍 교육을 위해 초보 프로그래머 교육에 도움이 될 수 있는 전략과 초 보 프로그래머가 프로그래밍 과정에서 부딪히는 문제점을을 관련 연구를 통해 제시하였다. 또한 프로그램 개발 과정에 기초한 프로그래밍 지식과 프로그래밍 전략의 학습을 위해 필요한 학습 요 소를 전문가 집단과 함께 추출해 보고, 더불어 프 로그래밍 교수학습 방법도 고민해 보았다. 본 논 문에서 제시한 프로그래밍 지식과 전략, 프로그래밍 교수학습 방법은 하나의 검증된 연구이기 보다는 앞으로 컴퓨터 교육 전공자들이 해결해야할 과제라고 생각한다. 더불어 본 연구를 통해 제시된 프로그래밍 교육에 관련된 다양한 이슈들을 좀 더 심도 깊은 현장 연구가 필요하다고 생각된다. 특히 프로그래밍 전략에 관한 연구와 초보 프로그래머들의 프로그래밍 오류에 대한 좀 더 심층 깊은 사례 연구, 다양한 방법이 적용된 프로그래밍 전략 관련 교재의 개발, 초보 프로그래머 교육에 적합한 교수학습 방법과 평가에 관한 연구또한 절실히 필요하다.

#### 참 고 문 헌

- [1] Kak, Avinash(2008), An essay on the state of how programming is currently taught in many universities, http://cobweb.ecn.purdue.edu/~kak/programming.pdf
- [2] Dehnadi & Bornat(2006). The camel has two humps. *PPIG 2006*. http://www.eis.mdx.ac.uk/research/PhDArea/saeed/paper1.pdf
- [3] 박영미·유관희 (2008). 초중등학교에서수학 교육체계와연계된컴퓨터프로그래밍교육과정 과교수방법. 한국콘텐츠학회논문지 8(1), 116-127.
- [4] 김갑수 (2009). 초등학생들을 위한 프로그래 밍 언어 교육 방법론. **서울교육대학교 한국** 초등교육, **19**(2), 135-152.
- [5] 배학진 외 (2009). 문제 중심 학습을 적용한 스크래치 프로그래밍 교수 학습 모형. 한국 컴퓨터교육학회논문지 12(3), 11-22.
- [6] 김병욱 외(2010). PBL 기반 프로그래밍 수업에서 학습양식에 따른 학습 동기 차이 분석을 통한 시사점 도출. 한국컴퓨터교육학회 논문지 13(5), 15-27.
- [7] 김미량 (2002). 컴퓨터 프로그래밍 교육에 적용 가능한 효과적 교수 방법의 탐색적 제안. 한국컴퓨터교육학회논문지 5(3), 1-9.
- [8] 문외식 (2005). 초등학생의 논리적 사고력 및 문제 해결 능력 향상을 위한 컴퓨터 프로그래밍 교육과정 모델 제안. 한국정보교육학회 논문지 9(4), 595-605.

- [9] 유관희 (2008). 초중등학교에서의 컴퓨터 프로그래밍 교육 과정. 한국컴퓨터교육학회지 **2**(1), 23-29.
- [10] 최해심 외 (2005). 두리틀 로봇 프로그래밍 일원화를 위한 로봇 객체 설계. 한국컴퓨터 교육학회논문지 8(6), 23-31.
- [11] 유인환·채재호 (2008). 로봇을활용한초등학 교프로그래밍교육방안. 한국정보교육학회는 문지12(3), 293-302.
- [12] 조성환 외 (2008). CPS에 기반한 스크래치 EPL이 문제해결력과 프로그래밍 태도에 미치는 효과. 한국정보교육학회논문지 12(1), 77-88.
- [13] 김수환 외 (2009). 개정된 정보교육과정에서 교육용프로그래밍언어의 교육적 적용 방안. 한국**컴퓨터교육학회논문지 12**(2), 23-31.
- [14] 김종훈·정원희 (2005). 창의성향상을위한프로그래밍교육방안. 제주교육대학교초등교육연구 제10집, 127-147.
- [15] 차승은 외(2009). 프로그래밍 교육과 필요성 의 인식 변화에 관한 연구. 한국컴퓨터교육 학회 논문지 12(1), 1-13.
- [16] Kirsti Ala-Mutka, Problems in Learning and Teaching Programming. http://www.cs.tut.fi/~edge/literature\_study.pdf
- [17] Perkins, D.N., & Martin, F. (1986). Fragile knowledge and neglected strategies in novice programmers. In E. Soloway & S. Iyengar (Eds.), *Empirical studies of programmers*, First Workshop, 213 229. Norwood, NJ: Ablex.
- [18] Soloway, E., & Spohrer, J.C. (Eds.). (1989). Studying the novice programmer. Hillsdale, NJ:Lawrence Erlbaum.
- [19] Davies, S.P. (1993). Models and theories of programming strategy. *International Journal of Man–Machine Studies*, 39, 237 267.
- [20] Winslow, L.E. (1996). Programming pedagogy
   A psychological overview. SIGCSE
  Bulletin 28, 17 22.
- [21] D. Storeya, F. D. Fracchiaa and H. A. Müllerb (1999). Cognitive design elements to support the construction of a mental model during software exploration, *Journal*

- of Systems and Software, 44(3), 171-185.
- [22] Du Boulay, B. (1989). Some difficulties of learning to program. In E. Soloway & J.C. Spohrer (Eds.). Studying the novice 283 - 299. programmer, Hillsdale, NJ: Lawrence Erlbaum.
- [23] Spohrer, J.C., & Soloway, E. (1989). Novice mistakes: Are the folk wisdoms correct? In Soloway & J.C. Spohrer Studying the novice programmer, 401 -416. Hillsdale, NJ: Lawrence Erlbaum.
- [24] Détienne, F. (1990). Expert programming knowledge: A schema based approach. In J.M. Hoc, T.R.G. Green, R. Samurc, ay, & Gillmore (Eds.), Psychology D.J. programming, 205 - 222, London: Academic Press.
- [25] Widowski, D., & Eyferth, K. (1986). Comprehending and recalling computer different programs of structural complexity semantic by experts novices. In H.P. Willumeit (Ed.), Human decision making and manual 267-275, Amsterdam: North-Holland Elsevier.
- [26] Brooks, R.E. (1983). Towards a theory of the comprehension of computer programs. International Journal of Man-Machine Studies, 18, 543 - -554.
- [27] Von Mayrhauser, A., & Vans, A.M. (1994). Program understanding: A survey (Tech. Rep. CS-94-120). Department of Computer Science, Colorado State University.
- [28] Shneiderman, B., & Mayer, R. (1979). Syntactic/semantic interactions in programmer model and experimental results. International Journal of Computer and Information Sciences, 8, 219-238.
- [29] Rist, R.S. (1995). Program structure and design. Cognitive Science, 19, 507-562.
- [30] Kolling (1999).The design an object-oriented environment and language for teaching, Phd thesis, University of Sydney.
- [31] McIver, Conway (1996). Seven Deadly Sins

- of Introductory Programming Language 1996 Design, Proceedings of the International Conference Software Engineering: Education and Practice.
- [32] Anthony Robins, et al. (2003). Learning and Teaching Programming: A Review and Discussion. Computer Science Education, 13(2), 137-172.



### 최 현 종

2001 한국교원대학교 컴퓨터교육과(교육학석사) 2005 한국교원대학교 컴퓨터교육과(교육학박사)

2006~현재 서원대학교 컴퓨터교육과 조교수 관심분야: 컴퓨터교육, Semantic Web E-Mail: blueland@seowon.ac.kr