

# 온라인 프로그래밍 개념학습 성취수준과 오류유형과의 관계 분석

김지선<sup>†</sup> · 김영식<sup>††</sup>

## 요 약

본 연구는 중·고등학생들의 온라인 프로그래밍 과제 수행결과에서 발생한 오류를 파악하여, 오류유형과 오류내용을 분류하고, 프로그래밍 개념학습 성취수준에 따른 오류 빈도의 차이와 성취수준과 오류유형과의 상관관계를 분석하여 향후 프로그래밍 교육에 대한 방향과 인지수준에 따른 교육방법을 제시하였다. 연구를 위해 88명의 학생들의 프로그래밍 과제 수행 결과를 가지고 문법오류, 논리오류, 코딩오류로 오류유형과 오류내용을 분류하고 분석하였다. 분석결과, 세 오류유형 중 논리오류의 발생비율이 69.3%로 가장 높았으며, 성취수준에 따른 오류 빈도의 차이에서는 성취수준 상, 중, 하 세 집단 간에 유의한 차이가 있었다. 성취수준과 오류유형과의 상관관계 분석 결과에서는 논리오류와 코딩오류에서 부적 상관관계를 보여, 성취수준이 높을수록 논리오류와 코딩오류를 적게 범함을 알 수 있었다. 오류유형간의 상관관계에서는 문법오류와 코딩오류간의 정적상관관계를 보였다.

**주제어** : 프로그래밍 개념 학습, 프로그래밍 오류, 오류유형

## The Analysis of Relationship between Academic Achievement Level of Concept Learning and Error Type in Online Programming Course

Jiseon Kim<sup>†</sup> · Yungsik Kim<sup>††</sup>

## ABSTRACT

This study has purpose on analyzing the error types which are identified after middle and high school students perform the online programming assignments and also has the purpose on the analysis of correlation between the frequency of error occurrence according to academic achievement level in programming concept learning and types of errors analyzed previously. For this study, the syntax, logical, and coding errors are analyzed from the performed results of programming research assignment for 88 students. Analyzed results show that the logical error has the highest occurrence rate of 69.3% among three types of errors, and it has been shown meaningful difference in the frequency of error occurrence between three achievement level groups of high, middle, and low. In the correlation analysis of achievement level and error types, it shows negative relationship between logical error and coding error, and therefore it can be concluded that as achievement level is higher, both logical and coding errors tend to occur less. In the correlation analysis in error types, it shows positive relationship between syntax error and coding error.

**Keywords** : Programming Concept Learning, Programming Error, Error Type

---

† 정 회 원: 한국교원대학교 박사과정  
 †† 종신회원: 한국교원대학교 컴퓨터교육과 교수(교신저자)  
 논문접수: 2014년 7월 1일, 심사완료: 2014년 8월 14일, 게재확정: 2014년 9월 12일  
 \* 본 논문은 한국교원대학교 2014년도 KNUE 학술연구비 지원을 받아 수행하였음

## 1. 서론

프로그래밍 교육의 목적은 알고리즘 사고력과 문제해결력을 키우는데 있으며, 이를 뒷받침할 수 있는 많은 연구들이 이루어져왔다[1]. 프로그래밍 교육을 통해 단순히 사고력이 향상되기 보다는 학습자가 얼마나 기본 지식을 갖고 있는지, 인지 발달 수준이 어느 정도인지에 따라 창의적 사고력과 논리적 사고력에 영향을 준다고 보고하였다[2]. 특히 프로그래밍 학습은 학습자의 계산적 사고와 논리적 사고력을 향상시키는 데 필수적인 교육이라고 할 수 있다. 그러나 창의적 문제해결 능력이 중요시 되는 지식기반의 교육에서 이러한 특징을 갖고 있는 프로그래밍 학습의 장점과 함께 학습자가 프로그래밍에 대한 개념지식의 이해와 습득뿐만 아니라, 문제해결과정을 컴퓨터가 이해할 수 있는 언어로 표현할 수 있는 능력이 별도로 요구된다. 즉, 컴퓨터 프로그래밍 언어 학습은 이해하기 힘든 문법, 텍스트형 코딩이라는 자연언어와 다른 차이점으로 인해 그 목적인 창의성과 문제해결능력을 향상시키는 알고리즘 학습을 효과적으로 달성하는 데 어려움이 있다[3].

그러나 프로그래밍 학습이 어렵다는 이유로 보다 단순한 패키지형 소프트웨어를 사용하여 흥미 위주와 도구 활용 중심의 학습으로 대체하기 보다는 프로그래밍 교육에서 보이는 오류 유형과 내용을 파악하여 학습자의 프로그래밍에 대한 인지수준을 이해하고, 오류 수정을 통한 학습자 자신의 행동에 대한 통찰이 이루어질 수 있도록 학습할 필요가 있다. 즉, 프로그래밍 과정과 마찬가지로 오류를 찾고 수정하는 디버깅 과정은 매우 중요한 의미를 가지고 있다[4]. 프로그래밍 학습에 관한 연구들을 살펴보면, 프로그래밍 능력과 논리적 사고력과의 관계에 대한 연구는 많이 이루어졌지만, 실제 프로그래밍을 하면서 겪는 개인별 어려움이나 수준별 오류 유형에 대한 연구는 많이 이루어지지 않았다.

따라서 본 연구에서는 온라인 학습 환경을 통해 이루어지는 중·고등학생 대상 프로그래밍 학습에서 나타나는 오류유형과 오류내용을 분석하여 온라인 프로그래밍 교육의 교수·학습 설계를 위한 기초자료를 마련하고자 한다. 또한 온라인

학습 시 즉각적인 피드백이나 오류 수정 등의 제약을 최소화하여 맞춤형 자기주도학습 지원이 필요하므로, 이를 위해서는 학습자의 프로그래밍 학습 능력에 대한 분석이 사전에 이루어져야 한다. 이러한 결과를 바탕으로 향후 수준별 온라인 프로그래밍 교육 방향과 인지수준에 따른 교육방법 등을 제시하고자 한다.

## 2. 이론적 배경

### 2.1 프로그래밍 교육

프로그래밍 교육은 교육의 대상이 되는 형식적인 언어 체계와 자연언어 체계와의 관계성을 파악하게 하는 것으로, 어떤 특정 체계에의 적용을 목적으로 만들어진 형식 언어 체계가 지니는 형식적 논리와 의미체계와 인간의 자연언어의 논리와 의미체계를 구조적으로 대응시킬 수 있도록 하는 것이다[13]. 또한 프로그래밍 교육은 컴퓨터로 실습하고 각자 터득해 가는 차이가 있기 때문에, 개인차이가 크게 발생하고, 이에 따라서 성취도 또한 달라진다[2]. 프로그래밍 교육은 문제해결력과 논리적 사고력 향상을 목적으로 하며, 각 학습자의 발달 단계에 맞는 프로그래밍 교육이 이루어질 수 있도록 해야 하며, 프로그래밍 언어 교육이 가지는 교육적 의미는 다음과 같이 정리할 수 있다[2, 5].

첫째, 프로그래밍 학습은 오류 수정 활동을 통해 논리적 사고력을 향상시킬 수 있다. 오류 수정은 여러 가지 어려움을 하나하나 처리해 감으로써 상대적으로 개선된 프로그램을 만들어 가는 것을 의미한다. 반복되는 오류수정 속에서 스스로 오류를 찾고 알고리즘을 분석하여 수정함으로써 논리적 사고력 향상에 도움을 주게 된다. 오류 수정을 위해서 교사는 일방적인 지적이나 일방적으로 가르쳐주는 것이 아닌 학습자가 알아갈 수 있도록 방향을 제시해주는 것이 좋다. 교사가 단순히 지적하거나 알려주는 것 보다 방향을 제시하여서, 프로그램 오류 수정 시 생각할 수 있게 한다면 논리적 사고력 향상에 도움을 주게 된다.

둘째, 프로그래밍 학습은 컴퓨터에 대한 이해를 돕고 스스로 문제를 해결할 수 있도록 도와준다.

주어진 문제에 대해서 스스로 해결해 나가게 되면 새롭게 부딪히는 문제에 도전을 할 수 있다. 단순히 응용 프로그램을 실행하는 방법만 가르치는 것에서는 기대할 수 없는 새로운 도전과 창조 의 의지를 가질 수 있도록 도움을 준다는 것이다.

## 2.2 프로그래밍 과정과 인지요소

프로그래밍 과정에서 요구되는 인지요소에 대해 백영균(1988)은 프로그래밍은 자세한 것에 관심을 기울여야 되고 방법적 사고과정이 요구되는 지적활동을 필요로 하며, 프로그래밍을 인지요구 활동이라고 하였다[6]. Kagan과 Douthat(1985)은 주어진 문제를 해결하기 위해 계획을 세우고, 오류를 고치는 디버깅은 프로그래밍 언어의 구문이나 어의를 기억하였다가 재생하는 것 이상의 인지적 기술이 필요하다[14].

Delbey와 Linn(1985)은 프로그래밍의 과정을 4 단계로 제시하였는데, 문제명시, 문제설계, 코딩, 오류정정으로 구분하였다. 각 과정에서 요구되는 인지요소에 대해 살펴보면 문제명시에서는 프로그래밍에 대한 어느 정도의 지식과 경험을, 문제설계 단계에서는 추상적 사고가 필요하다고 하였다[15]. 이는 특정한 문제를 해결하기 위해서는 여러 가지 서로 다른 방법이 있기 때문이며, Sheil(1981a)는 이 과정을 절차적 사고라고 하였다. 코딩 단계에서는 높은 수준의 인지기술이 필요하지 않지만, 사용하는 프로그래밍 언어의 구문과 문법을 기억해야 하며, 많은 시간과 노력이 필요한 오류수정 단계에서는 합리적 사고, 창의적 사고 및 확산적 사고가 요구된다고 하였다[16]. 즉, 특정한 문제에 대해 하나 이상의 해결책이 있다는 사실이 학생들의 창의적 사고와 확산적 사고를 돕는다고 할 수 있다[15].

## 2.3 프로그래밍 오류 유형

프로그래밍 과정동안 발생하는 오류는 프로그래밍 심리학, 소프트웨어 공학, 프로그래밍 교육, 지능형 교수 시스템 분야의 연구자들에게 중요한 주제로 인식되어 왔다[7]. 프로그래밍 오류는 학습자의 능력 부족에서 발생하는 것으로, 잘못된 학

습 방법이 체계적이고 반복적으로 나타나는 것이라고 말하며 결국 오류가 학습과정의 어느 특정 시기에 학습자가 사용하는 학습 체계를 보여주는 증거물이라고 할 수 있으므로 학습자의 학습 행동에 대한 오류 분석이 아주 중요하다[7].

프로그래밍 오류를 하나의 분류 체계로 구분하는 것은 어려우며, 많은 연구자들의 관점에 따라서 분류되었다. Murray(1987)는 프로그래밍 오류를 지식유형에 기초하여 5가지로 분류하였다[17]. 첫째, 잘못된 문제 해석의 오류(misinterpretation)로 문제의 핵심을 파악하지 못해서 발생하며, 잘못된 목표를 설정하게 되고, 그 결과는 엉뚱한 프로그램이 되게 된다. 둘째는 전략적 오류(strategic error)로 문제해결을 위한 방법의 세부 절차를 계획하는 과정에서의 오류를 의미한다. 즉, 잘못된 절차를 포함했거나, 필요한 절차들을 누락했거나, 발생 가능한 대안 상황들에 대해 고려를 하지 않은 경우 등이 포함된다. 초보자의 경우, 전략적 오류를 통해 저지르는 실수는 잘못된 전제를 통한 오류를 범하게 된다. 세 번째는 번역상의 오류로 자연언어의 논리를 이용해 계획을 세우고 세부 방법 절차들을 계획한 후 프로그래밍 언어의 논리와 규칙을 이용해 그 계획을 번역하는 과정에서 발생하는 오류이다. 특히 이 오류는 프로그래밍 언어의 논리와 지식의 부족으로 발생하는 오류로 부적절한 명령어 사용, 불필요한 루프 사용, 잘못된 코드 블록 배치 등이 이에 해당된다. 네 번째는 통사적 오류로 프로그래밍 언어에 대한 지식이 잘못되어 저지르는 광범위한 오류로, 변수 선언 누락, 다른 변수 유형 사용 등 일반적인 문법 규칙을 위배하는 오류이다. 다섯 번째 오류는 코딩오류로 타이핑 잘못으로 인한 오타자, 계산식에서의 연산자 누락 등의 실수에 의한 오류들을 의미한다.

프로그래밍 오류에 대한 국내 연구자들의 오류 분류는 다음과 같다. 송종수(1998)는 자동오류 분석기에 관한 연구에서 초보자들이 저지르기 쉬운 오류를 전략적 오류, 함수적 오류, 구현상의 오류로 분류하였다[8]. 정란(1996)은 오류의 유형을 프로그래밍 언어가 제공하는 기능을 제대로 이해하지 못하여 생기는 구문-기반 bug, 프로그램 내의 정보 흐름을 파악하지 못하면서 계획을 적절하게

통합시키지 못해 생기는 구문-기반이 아닌 bug, 그리고 구문-기반인 듯 하면서도 반드시 구문-기반이 아닌 bug로 구분하였다[9]. 김은옥(2003)은 ICT활용 오류 유형을 기능 혼동 오류, 개념 혼동 오류, 인터페이스 해석 장애 오류, 학습자의 심리적 불안으로 인한 오류, 학습자 성격 유형에 의한 오류로 분류하였으며[10], 로봇 프로그래밍 학습에서 오류유형을 분류한 문외식(2008)은 로봇 프로그래밍에서 오류 유형을 논리 오류, 데이터정의 및 취급오류, 인터페이스 오류, 기타 오류로 분류하였고[3], 남재원(2010)은 문법오류, 논리오류, 코딩오류, 로봇 동작 오류로 분류하였다[11]. 이상의 기존 연구의 프로그래밍 오류를 정리해보면 <표 1>과 같다.

<표 1> 기존 연구의 오류 유형 분류

연구자	분야	오류유형
Murray (1987)	프로그래밍 언어	잘못된 문제해석의 오류
		전략적 오류
		번역상의 오류
		통사적 오류
		코딩 오류
송종수 (1998)	프로그래밍 언어	전략적 오류
		함수적 오류
		구현상의 오류
정란 (1996)	프로그래밍 언어	구문-기반 bug
		구문-기반이 아닌 bug
		구문-기반인 듯 하면서도 반드시 구문-기반이 아닌 bug
		구문-기반이 아닌 bug
김은옥 (2003)	ICT 활용	기능-혼동 오류
		개념-혼동 오류
		인터페이스 해석 장애 오류
		학습자의 심리적 불안으로 인한 오류
		학습자 성격 유형에 의한 오류
문외식 (2008)	로봇 프로그래밍	논리오류
		데이터정의 및 취급오류
		인터페이스 오류
		기타 오류
남재원 (2009)	로봇 프로그래밍	문법오류
		논리오류
		코딩오류
		로봇 동작 오류

2.4 프로그래밍 오류 분석

오류 분석에 대해 김은옥(2003)은 학습자의 문제점을 진단하기 위하여 학습자가 범하는 오류를 수집하여 분류하고 분석하며 그 빈도에 따라 난

이도를 추정하는 것으로 오류의 본질이 발견되는 명확한 과정이라고 하였다[15]. 또한 백광운(2002)은 오류를 진단, 분석, 수정하는 오류 분석 단계를 5단계로 구분하였으며, 각 단계는 다음과 같다 [12].

첫 번째 단계는 학습자들의 여러 가지 오류 활동들을 선정하는 단계로 이 때 학습자의 특징 즉 나이, 발전 단계, 활동 동기 등을 고려해야 한다.

두 번째 단계는 오류들을 정의하는 단계로, 명백한 오류와 무심결에 범하는 오류로 구분하며, 차후에 수정할만한 가치가 있는 오류인지, 무시해도 좋은 오류인지를 확인하는 단계이다.

세 번째 단계는 오류들을 일정한 기준에 따라 분류하는 단계이다.

네 번째 단계는 어떤 오류들이 발생했는지 내용을 기술하고 평가하는 단계이다.

마지막 다섯 번째 단계는 피드백을 통해 학습자들이 오류를 직접 확인하고 수정하며 예방하는 단계이다.

3. 연구방법 및 절차

본 연구에서는 중등 학생들을 대상으로 온라인 프로그래밍 개념학습 12차시와 탐구학습 4차시를 실시한 후, 탐구학습의 프로그래밍 과제를 수행한 결과에서 발생한 오류들을 수집하였으며, 제안한 오류유형별로 분류하였다. 또한 개념학습의 성취 수준에 따른 오류빈도의 차이와 오류유형과의 상관관계를 분석하였다.

본 연구의 연구문제는 다음과 같다.

- 1) 오류유형 중 발생 비율이 가장 높은 오류 유형과 오류 내용은 무엇인가?
- 2) 프로그래밍 개념학습의 성취수준에 따른 오류빈도의 차이는 무엇인가?
- 3) 프로그래밍 개념학습 성취 수준과 오류 유형과의 상관관계는 어떠한가?

3.1 연구대상

본 연구는 K대학교 영재교육센터에 온라인 정보 C언어 강좌를 수강한 중·고등학생 중 프로그래

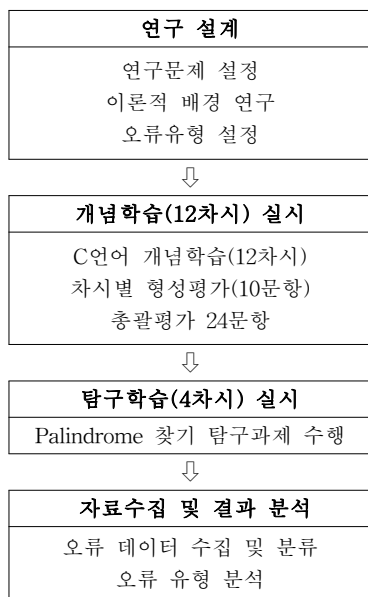
밍 과제를 끝까지 수행한 88명을 대상으로 진행하였다. 88명 중 성별로는 남학생이 69명으로 전체 학생의 78.4%, 여학생이 19명으로 전체학생의 21.6%의 비율을 차지한다. 학교급에 따른 인원은 중학생이 37명으로 전체학생의 42.0%, 고등학생이 51명으로 전체학생의 58.0%로 <표 2>와 같다.

<표 2> 연구 대상 표본

구분	중학생		고등학생		계	
	빈도	비율	빈도	비율	빈도	비율
남	31	35.2%	38	43.2%	69	78.4%
여	6	6.8%	13	14.8%	19	21.6%
계	37	42.0%	51	58.0%	88	100.0%

### 3.2 연구절차

본 연구는 오류유형을 분석하기 위해 3단계로 걸쳐 실시되었다. 첫 번째 단계에서는 C언어 프로그래밍에 대한 개념학습 12차시 학습이 이루어졌으며, 두 번째 단계에서는 실제로 프로그래밍 과제가 주어지는 탐구학습으로 4차시로 개념학습과 탐구학습은 2013년 9월부터 2013년 12월까지 실시되었다. 세 번째 단계에서는 오류데이터를 수집 및 유형별 분류하고 분석하였으며, 절차는 [그림 1]과 같다.



[그림 1] 연구절차

### 3.3 오류유형 및 오류내용

본 연구에서는 이론연구에서 남재원이 분류한 오류유형을 바탕으로 프로그래밍 오류를 <표 3>과 같이 3가지 유형 즉, 문법오류, 논리오류, 코딩오류로 분류하고 각 유형의 오류내용을 분석하였다.

<표 3> 오류 유형 및 오류 내용

유형	내용	예시
문법 오류	일반적인 문법 규칙 위배	명령어 누락, 문법에 어긋난 조건문, main 함수 사용 오류 등
	변수에 대한 잘못된 개념	변수 선언 누락, 다른 변수 유형 사용, 변수 자료형과 맞지 않은 자료 입력
	헤더파일 누락	stdio.h, string.h 누락
논리 오류	문제해석 오류	잘못된 문제 해석으로 인한 잘못된 결과
	불필요한 절차 포함	불필요한 함수, 루프 사용 등
	필요한 절차 누락	종료 조건 누락 등
코딩 오류	문제해결 전략 부족	반복문의 잘못된 설계, 부적절한 배열 크기 선언
	실수로 인한 오타자	오타자, ')'누락, ';' 누락 등 계산식에서의 연산자 누락

각 오류유형에 대해 살펴보면, 문법오류는 프로그래밍 언어가 가지고 있는 규칙이나 문법을 어겼을 때 발생하는 오류로 컴파일할 때 발견할 수 있으며, 디버깅하여 해결할 수 있는 오류이다. 본 논문에서는 기존연구의 오류내용 분류를 근거로 하여, 일반적인 문법 규칙 위배, 변수에 대한 잘못된 개념 및 헤더파일 누락을 세부 오류내용으로 선정하였다. 논리오류는 문법적으로는 전혀 문제가 없는 오류로 실행결과에는 영향을 끼치지 않으나, 문제에서 요구하는 결과가 아닌 다른 결과를 초래하는 오류이다. 논리오류는 크게 4가지로 분류하였는데, 문제해석 오류, 불필요한 절차를 포함했거나, 필요한 절차를 누락한 경우 및 문제해결 전략 부족에서는 발생한 오류이다. 마지막으로 코딩오류는 문법오류와 일치할 수 있으나, 오타자에서 발생하는 오류로 ')', ';'의 누락 등을 조사하였다. 또한 오류유형을 분류하는 과정에서 프로그래밍 오류판정에 대한 오차를 최소화하기

위해, 학생들이 작성한 프로그램 소스를 3차에 걸쳐 컴파일하고 최종적으로 오류유형별로 분류하였다.

### 3.4 자료 분석 방법

본 연구에서 사용한 분석방법은 다음과 같다.

첫째, 프로그래밍 학습에서 발생하는 오류유형을 살펴보기 위해, 빈도, 비율, 평균, 표준편차와 같은 기술통계방법을 사용하였다. 둘째, 개념학습 성취수준에 따른 오류빈도의 차이를 분석하기 위해 일원분산분석 (one-way ANOVA)을 사용하였고, 오류유형과의 상관관계를 탐색하기 위해 단순상관관계분석 (Pearson's correlation analysis)을 사용하였다. 셋째, 모든 통계분석은 SPSS를 사용하였다.

## 4. 연구결과

### 4.1 학교급간 동질성 검사 결과

본 연구에서는 중학생과 고등학생의 인지적 수준의 차이가 없다는 가정 하에 성취수준에 따른 오류빈도나 오류유형의 차이를 분석하였다. 이를 검증하기 학교급에 따른 성취수준의 차이검정을 독립표본 t-검정을 실시하였다. Levene의 등분산 가정이 유의확률 .052로 두 집단의 분산이 같다는 영가설을 기각하지 않아, 등분산 가정이 충족되었고, <표 3> 과 같이 t-검정 결과, t값은 .167, 유의확률은 .868로  $p < .05$  이다. 따라서 학교급에 따른 두 집단 간에는 차이가 없는 것으로 나타났다. 따라서 학교급에 따른 전체 학생들은 동질 집단임이 밝혀졌다.

<표 4> 학교급에 따른 성취수준 차이

학교급	N	M	SD	t	p
중학생	37	16.83	4.74	.167	.868
고등학생	51	16.68	3.69		

### 4.1 오류데이터 수집 및 분석 결과

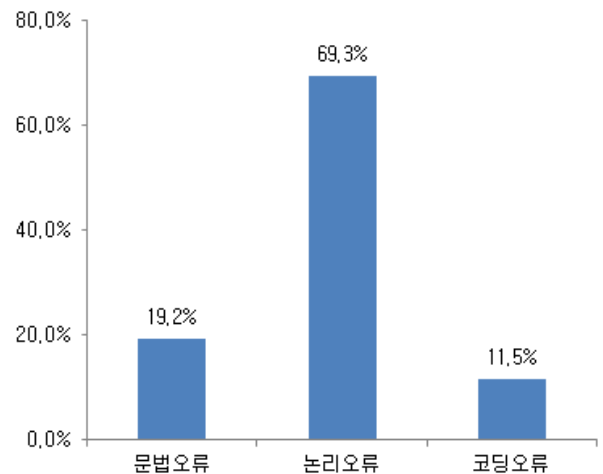
탐구학습 과제 결과를 통해 수집된 오류 데이

터는 <표 3>에서 제안한 오류유형별로 분류하였으며, 결과는 <표 5>와 같다. 3개의 오류유형 중 발생 비율이 가장 높은 오류는 논리오류로 69.3%를 차지하며, 다음은 문법오류는 19.3%, 그리고 코딩오류가 11.4%로 나타났다. 가장 많이 발생한 논리오류 내에서 발생비율이 가장 높은 오류내용은 필요한 절차 누락(27.1%)이었고, 그 다음은 문제해결 전략 부족(24.7%)으로 나타났다.

<표 5> 오류유형별 오류분류 결과

유형	내용	빈도(%)	유형별 빈도(%)
문법오류	일반적인 문법 규칙 위배	16(9.6%)	32(19.2%)
	잘못된 변수 개념	11(6.6%)	
	헤더파일 누락	5(3.0%)	
논리오류	문제해석 능력 부족	19(11.5%)	115(69.3%)
	불필요한 절차 포함	10(6.0%)	
	필요한 절차 누락	45(27.1%)	
	문제해결 전략 부족	41(24.7%)	
코딩오류	실수로 인한 오타자	19(11.5%)	19(11.5%)

오류 유형별 발생 비율은 [그림 2]와 같다.



[그림 2] 오류 유형별 발생 비율

### 4.3 성취 수준에 따른 오류빈도 차이

개념학습 성취 수준에 따른 오류유형의 특징을

알아보기 위해, 성취수준의 지표를 개념학습의 형성평가 점수와 총괄평가 점수의 평균을 사용하였다. 그리고 평균 점수에 따라 상, 중, 하 3그룹으로 구분하였으며. ‘상’인 그룹은 전체 학생 중 상위 25%이내, ‘중’에 해당하는 학생은 25%~75%, ‘하’는 하위 25%에 해당하는 학생으로 하였다. <표 6>은 개념학습 성취 수준에 따른 오류의 발생빈도와 비율을 나타낸 것이다. 그룹별 오류유형 발생의 특징은 공통적으로 논리오류를 많이 범했으며, 개념학습 성취수준이 ‘상’ 그룹의 학생들은 문법오류와 코딩오류에서 낮은 비율을 보였다. 또한, ‘중’에 속한 학생들의 경우, 코딩오류 전체 발생 비율 중 66.7%를 차지하였다. ‘하’에 속한 학생들 역시 논리오류에서 발생비율이 높음을 알 수 있으며, 문제해석 능력 부족 오류내용에서 다른 그룹 중 가장 높은 비율을 보였다.

<표 6> 개념학습 수준별 오류발생 차이(N=88)

구분	상 (N=22)	중 (N=44)	하 (N=22)
	빈도(비율)	빈도(비율)	빈도(비율)
문법오류	3(10.7%)	17(20.5%)	12(21.8%)
논리오류	24(85.7%)	57(68.7%)	34(61.8%)
코딩오류	1(3.6%)	9(10.8%)	9(16.4%)

개념학습 성취수준에 따른 전체 오류빈도의 평균과 표준편차는 <표 7>과 같다.

<표 7> 개념학습 성취수준별 오류발생 결과

	상	중	하	전체
평균	1.28	1.88	2.5	1.88
표준편차	1.16	1.16	1.3	1.25
사례수	22	44	22	88

상위 그룹의 학생들의 오류발생 빈도 평균은 1.28, 표준편차는 1.16이고, 중위 그룹의 학생들은 평균 1.88, 표준편차 1.16이다. 하위 그룹에 속한 학생들의 오류발생 빈도 평균은 2.5, 표준편차는 1.3이다. 개념학습 성취수준에 따라 오류에 차이가 있는지 알아보기 위하여 일원분산분석을 실시

하였으며, 결과는 <표 8>과 같다.

<표 8> 개념학습 성취수준에 따른 오류발생에 대한 일원분산분석 결과

	SS	df	평균제곱	F	p
성취수준	16.57	2	8.284	5.95	.004
오차	118.30	85	1.392		
합계	134.86	87			

\* $p < .05$

세 집단에 대한 F통계 값이 5.95, 유의확률은 .004로서 유의수준 .05에서 개념학습 성취수준에 따른 학생들의 오류발생에 유의한 차이가 있었다.

#### 4.3 성취수준과 오류유형과의 상관관계

성취수준과 오류유형과의 상관분석을 실시한 결과가 <표 9>에 제시되어 있다. <표 9>에서 보여주는 결과와 같이 개념학습 성취수준과 논리오류( $r = -.211, p < .05$ ), 코딩오류( $r = -.265, p < .01$ )와는 부적의 상관관계를 가지는 것으로 나타났다. 즉, 개념학습 성취수준이 높을수록 논리오류와 코딩오류 빈도는 낮음을 알 수 있었다.

오류유형간의 상관관계를 살펴보면, 문법오류와 코딩오류( $r = .299, p < .01$ ) 간에 정적인 상관관계를 보였다. 즉, 문법오류의 빈도가 높으면, 코딩오류의 빈도도 높은 것을 알 수 있다.

<표 9> 성취수준과 오류유형과의 상관관계

	성취수준	문법오류	논리오류	코딩오류
성취수준	1			
문법오류	-.193	1		
논리오류	-.211*	-.101	1	
코딩오류	-.265**	.299**	-.203	1

\* $p < .05$ , \*\* $p < .01$

## 5. 결론 및 제언

본 연구에서는 중·고등학생을 대상으로 프로그래밍에 대한 개념학습과 탐구학습을 실시한 후, 탐구과제를 수행한 결과에서 프로그래밍 오류유형과 개념학습 성취수준이 어떠한 상관관계를 가지며, 성취수준에 따른 오류빈도와 차이를 분석하고자 하였다. 연구 결과, 학생들이 가장 많은 오류를 범하는 유형은 논리오류로 그 중에서도 필요한 절차를 누락하거나 문제해결전략부족으로 인한 오류가 높았다. 실제로 학생들의 프로그래밍 소스 코드를 분석해보면, 종료조건을 제대로 하지 않아, 종료가 되지 않거나, 한번만 실행 후에 바로 종료하는 프로그램으로 작성하였다. 그 다음에 많이 발생하는 논리오류 중 문제해결전략 부족으로 인한 오류 역시 반복문의 동작범위나, 배열의 크기를 잘못 설정하는 경우가 많았다.

개념학습의 성취수준에 따른 오류발생의 차이를 분석한 결과, F통계 값이 5.95, 유의확률은 0.5 수준에서 .004로 유의미한 차이가 있음을 알 수 있었다. 이는 성취수준이 높을수록 오류발생 비율이 낮음을 의미한다. 성취수준과 오류유형과의 상관관계를 분석한 결과, 성취수준과 논리오류, 코딩오류간에는 부적의 상관관계를 보여, 성취수준이 높을수록 논리오류나 코딩오류는 적게 범하는 것을 알 수 있었다. 각 오류유형간의 상관관계는 문법오류와 코딩오류간의 정적인 상관관계가 있었으며, 이는 문법오류의 발생빈도가 높은 학생들이 코딩오류도 자주 범하는 것을 알 수 있었다.

이상의 오류유형 분석 결과는 다음과 같은 의미와 프로그래밍 학습 방향을 제시한다. 첫째, 프로그래밍 언어 학습에 있어서 가장 발생빈도가 높은 논리오류 중 필요한 절차 누락과 문제해결전략 부족을 줄이기 위한 방법으로 논리적 사고와 문제해결전략에 대한 교육이 필요하다. 구체적인 교육방법으로는 문제를 해결하기 앞서, 문제를 정확히 인지하고 문제해결방법, 즉 알고리즘을 설계하는 과정이 선행되어야 한다. 이를 위해 플로우차트와 같은 방법을 이용하여 문제해결 절차를 기술하고, 실제 프로그래밍 과정과 매칭하는 방법을 들 수 있다. 즉, 프로그램에서 오류가 발생하였을 때 단순히 코딩 수정이 아닌 알고리즘 설계

를 점검하면서 학생 스스로 문제 해결에 대한 논리적인 접근 방법을 따를 수 있도록 이러한 과정이 전체 프로그래밍 학습에 포함되어야 한다.

둘째, 성취수준 중, 하 수준에서 많이 발생하고 있는 문법오류와 코딩오류를 줄이기 위해서는 프로그래밍 기초 학습이 중요하다. 본 연구에 참여한 학생들에게서 자주 보였던 C언어 기본 문법인 '='와 '=='의 개념 혼동이나 '%d'와 같은 식별자의 의미 파악부재에서 오는 오류 등은 기초 프로그래밍 언어 문법에 대한 반복적인 학습이 필요함을 나타내고 있다. 이를 위해 주요 문법에 대한 다양한 예제 프로그래밍 학습이나 기초에서 심화에 이르는 나이도별 문법 중심의 프로그래밍 학습을 통해 기초 문법을 익히고 활용 능력을 향상시킬 수 있도록 해야 한다.

셋째는 온라인 프로그래밍 환경에서의 보다 체계적인 튜터 지원을 통해 튜터는 학습지원자 역할뿐만 아니라 학습촉진자의 역할을 수행해야 한다. 구체적인 지원방법으로는 학생들이 프로그래밍 언어 과제를 수행하면서 겪는 어려움과 문제점을 성찰일지를 통해 기술하고, 튜터들은 학생들이 작성한 프로그래밍 오류 성찰에 대해 개별 피드백을 제시하여 같은 실수를 반복하지 않도록 해야 한다. 특히 대부분의 학생들이 범하고 있는 논리오류의 경우 프로그램 실행 결과에는 영향을 주지 않기 때문에 온라인 프로그래밍 학습에서는 차별화된 피드백 튜터링이 더욱 요구된다.

본 연구는 중·고등학생들이 프로그래밍을 하면서 실제로 겪는 오류의 유형과 내용을 파악하고, 프로그래밍 학습 방향을 제시한다는 점에서 의미를 가질 수 있다. 향후에는 프로그래밍 과정에서 발생하는 어려움이나 오류에 대한 성찰일지를 작성하고, 이에 대한 튜터 피드백을 제공하는 개인 맞춤형 프로그래밍 학습지원에 대한 연구가 필요하다.

## 참고 문헌

- [1] 김수환, 이원규, 김현철(2009). 개정된 정보교육과정에서 교육용 프로그래밍언어의 교육적 적용방안. 한국컴퓨터교육학회 논문지. 12(2), 23-31.



[2] 유병진(2012). 정보영재와 일반학생의 프로그래밍 문제해결과정 비교분석. 석사학위 논문, 고려대학교.

[3] 문외식(2008). 로봇 프로그래밍 학습에서 문제해결력에 영향을 미치는 오류 요소, 한국정보교육학회지논문지, 12(2), 195-202.

[4] 이유순(1995). 논리적 사고력 및 문제해결력 신장을 위한 컴퓨터 프로그래밍 교육. 석사학위 논문, 이화여자대학교 교육대학원.

[5] 강성원, 이애정, 이재호(2003). 초등정보과학 영재용 프로그래밍 교육(비주얼 베이직을 이용한 접근). 한국정보교육학회논문지, 7(3), 363-371.

[6] 백영균(1988). 컴퓨터 프로그래밍에 대한 심리학적 접근, Logo를 중심으로. 교육공학연구, 4(1), 145-165.

[7] 이정모, 이건효(1998). 초보자의 C언어 학습 과정에 대한 인지심리학적 분석 연구:프로그래밍 학습과정 동안의 은유 사용의 효과. 한국인지과학회 연구보고서. 1-20.

[8] 송종수(1998). 플랜정합과 프로그램 실행에 의한 초보자 프로그램 자동 오류분석에 관한 연구. 한국과학기술원 박사학위 논문.

[9] 정란(1996). 초보자 프로그래밍 과정의 특성 분석과 지원 방안. 삼척대학교 논문집 29(1), 387-411.

[10] 김은옥(2003). 아동의 ICT 활용 오류 유형 분석 및 처치방안. 석사학위 논문, 대구교육대학교.

[11] 남재원(2010). 로봇 프로그래밍 과정에서 발생하는 아동의 오류 유형 분석과 지원도구 개발. 석사학위논문, 대구교육대학교 교육대학원.

[12] 백광운(2002). ICT소양교육 학습을 위한 오류 분석-피드백 시스템. 석사학위 논문, 서울교육대학교 교육대학원.

[13] Gentner, D. and Jezrioski, M(1983). The shift from metaphor to analogy in Western science. In A. Ortony(Ed.), Metaphor and Thought. New York; Cambridge University Press.

[14] Kagan, Dona M. and Douthat, John M.(1985). Personality and Learning FORTRAN. International Journal of

Man-Machine Studies, 22, 395-402

[15] Delbey, John and Linn, Marcia C.(1985). The Demands and Requirements of Computer Programming: A Literature Review. Journal of Educational Computing Research. 1(3), 253-274.

[16] Sheli, B.A.(1981a). The Psychological Study of Programming, Computing Surveys. 13(1), 101-120.

[17] Murray, W. R.(1987). Automatic program debugging for intelligent tutoring systems. Computational Intelligence. 3, 1-16.



### 김 지 선

2001 한밭대학교  
컴퓨터공학과(공학사)

2003 충북대학교  
컴퓨터학과(이학석사)

2011~현재 한국교원대학교 영재교육통합과정  
박사과정

관심분야: 정보영재, 온라인 교육, 교육과정  
E-Mail: jskim315@kaist.ac.kr



### 김 영 식

1982 서울대학교  
전기공학과(공학사)

1987 노스캐롤라이나주립대학교  
전기 및 컴퓨터공학(공학석사)

1993 노스캐롤라이나주립대학교  
전기 및 컴퓨터공학(공학박사)

1993~1994 한국전자통신연구소 선임연구원  
1995~1996 한국전자통신연구소 위촉연구원  
1996~1998 한국전자통신연구소 초빙연구원  
1994~현재 한국교원대학교 컴퓨터교육과 교수

관심분야: 컴퓨터교육, e-Learning, ITS  
E-Mail: kimys@knue.ac.kr