

1. 서론

21세기 정보화 사회를 이끌기 위한 교육목표는 핵심적인 학습 능력을 연구하고 창의적 문제해결을 통해 미래사회와 국가 발전에 필요한 IT 융·복합 시대를 선도할 수 있는 핵심 인력을 발굴하고 육성하는데 있다. 정부의 창조경제 추진 등과 맞물려 교육 분야에서도 창의력 발달교육에 대한 관심이 높아지고 있다[1]. 글로벌 인재포럼에서는 창의적이고 발전적인 인재를 누가, 어디서, 어떻게 키워내느냐 하는 인재육성이 교육과 사회, 경제, 산업 등 모든 분야에서 가장 중요한 핵심 과제라고 제시하였다[2].

창의력이란 독창적이고 가치 있는 아이디어나 작품을 생성해낼 수 있는 능력으로 정의되거나[3], 어떤 개인이나 집단의 특성이 창의적인 과정을 거치면서 새롭고 어떤 가치 있는 결과를 만들어 내는 것이라고도 정의된다[4]. 창의력에 대한 관심이 고조되면서 많은 학자들은 창의력을 포함하여 이와 관련된 다양한 사고력들을 함께 구분하는 연구를 진행하여 왔다. 김영정의 연구[5]는 논리적 사고와 창의적 사고 영역을 제시하였는데, 논리적 사고 안에 기초적 사고, 분석적 사고, 추론적 사고, 종합적 사고를 포함시켰고, 창의적 사고에 대한적 사고, 발산적 사고, 상징적 사고를 포함시켰다. 그의 연구는 사고의 다차원성을 종합적인 틀에서 분석하였지만 정작 추상적 사고(abstract thinking)는 고려하지 않았다.

Glick과 Wapner의 연구[6]에 따르면 8~18세의 청소년들은 추상적인 언어적 자료의 제시보다는 그림과 같은 시각적인 구체적 사례를 제시했을 때 추론문제를 더 잘 해결하는 것으로 드러났다. 이와 같이 문제의 언어 자료 제시는 추상적인 것으로, 문제의 시각적 자료 제시는 구체적인 것으로 간주되어 왔다. Trope과 Liberman은 심리적 거리(psychological distance)가 멀수록 하위 수준의 구체적 해석(low level construal)보다는 상위 수준의 추상적 해석(high level construal)을 하게 된다는 해석수준이론(Construal Level Theory)을 발표하였다[7]. 김병원의 연구[8]는 추상적 이유는 창의기능의 발로이고 구체적 이유는 연상 사고의 발로라는 Runco의 이론[9]을 기반으로 과학적 사

고에서의 추상성과 창의성의 관계를 기술하였다. 이 연구는 추상적인 이유가 구체적인 이유보다 비판적 사고로서 반론에 강하다는 것을 입증하였다.

Kramer[10]는 30년 이상의 프로그래밍 수업에서 축적된 경험을 바탕으로 추상적 사고가 컴퓨터 프로그래밍 학습능력 향상에 중요하다고 주장하였다. 그는 추상의 정의와 역할을 제시하였고, 추상화 기술이 적절한 모델과 설계, 구현을 하는데 핵심적이라고 설명하였다[10]. 또한 그는 추상적 사고가 프로그램을 분석하고 프로그램을 작성하기 위한 형식적 모델들을 생각해내는데 필수적이라고 제시하였다. 그러나 그는 추상적 사고가 프로그래밍 학습능력에 어떠한 영향을 주는지는 기술하지 않았다.

추상적 사고의 중요성을 강조한 기존 연구들은 추상적 사고의 정의와 특성을 분석하는 데 초점을 두었을 뿐, 추상적 사고가 문제를 해결하는데 어떤 방식으로 영향을 미치는지 구체적인 관계를 조사한 것은 드물다. 특히, 컴퓨팅 사고력에서도 추상화에 대한 중요성을 강조하지만[11], 청소년들의 추상적 사고력 증진을 위한 방법론적인 연구 역시 드물다.

추상적 사고 이외에도 비전문가들이 프로그래밍 언어를 성공적으로 학습하기 위한 요인 분석에 관한 연구가 존재한다. 그 중에서 경험이 프로그램을 성공적으로 학습하기 위한 요소로 간주되고 있다[12]. 프로그래밍 언어에 대한 경험은 결국 프로그래밍 언어 친밀성이라 할 수 있다.

본 연구에서는 추상적 사고와 언어 친밀성, 학업 성취도가 프로그램 이해력과 어떤 관계를 가지며 어떻게 영향을 미치는지 분석하고자 한다. 이를 통해 추상적 사고는 어떤 경우 프로그램 이해에 도움을 주며, 프로그램 이해력을 높이기 위해 어떤 방법으로 추상적 사고 수준을 높일 수 있는지 제언하고자 한다. 즉, 본 연구의 목적은 궁극적으로는 실증 분석을 통해 프로그램 이해력에 도움을 주도록 추상적 사고력을 증진시킬 수 있는 교육적 방법에 대한 시사점을 제공하고자 한다.

이를 위하여 첫째, 본 연구에서는 일반계 고등학생을 대상으로 추상적 사고 수준을 측정하고 학생들이 본 연구를 위한 설문에 응답하기 직전

까지 배운 프로그래밍 언어의 종류를 조사한다. 둘째, C 언어를 학습한 고등학생과 스크래치(Scratch) 언어를 학습한 고등학생들에 대해 작성된 프로그램의 결과를 어느 정도 정확하게 예측하는지 프로그램 이해력과 학업성취도를 측정하였다. 본 연구는 다양한 프로그래밍 요소들 중에서 기본적인 제어구조(순차, 선택, 반복)를 중심으로 학생들의 프로그램 이해력에 초점을 두었다. 컴퓨터 프로그래밍 초보자인 고등학생들을 중심으로 하기 때문에 여러 가지 프로그래밍 능력 중에서 보다 기본적인 능력인 프로그램의 제어구조에 대한 이해력을 우선시 하였다. 셋째, 프로그램 이해력과 학업성취도간의 관계 및 추상적 사고 수준, 프로그래밍 언어친밀성간의 관계를 분석하여 프로그램 이해력 증진을 위해서 추상적 사고력이 유의미하게 발휘되는 경우를 파악하고 상호작용 효과분석을 통해 비전문가인 일반계 고등학생들의 성공적인 프로그래밍 교육에 도움을 주고자 교육적 제언을 진술하였다.

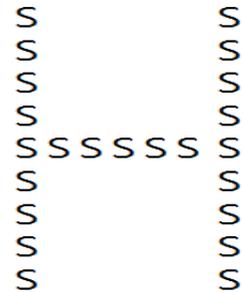
2. 이론적 배경

2.1 추상적 사고의 개념과 특성

추상(abstract)과 추상화(abstraction)에 대한 여러 분야에서의 연구는 방대하다[13][14][15][16]. Burgoon의 추상화에 대한 한 연구[14]에서 여러 학자들이 제시한 정의들의 공통점을 추출하였다. 첫 번째 공통점이 추상은 정보와 형식 판단, 행동 규제를 학습하는 방법에 영향을 준다는 사실이었다. 두 번째는 추상화는 물체의 불변하는 핵심속성을 정의하는 과정이었다. 이 연구[13]에서는 추상화를 측정하는 연구들을 조사하여 제시하였는데 100회 이상 측정도구로 인용된 연구가 <그림 1>에서와 같은 Navon의 계층적 글자(hierarchical letters) [15]와 Vallacher와 Wagner의 행동정체성 이론(Behavioral Identification Theory)에 관한 것이었다[13][16].

<그림 1>의 글자는 지역적으로는 알파벳 S로 보이지만 전체적으로는 알파벳 H로 보인다. 해석 수준이론에 관한 연구에서는 응답자들로 하여금 알파벳 S를 읽게 했을 때는 구체적 정보처리가

높은 반면에 알파벳 H로 읽게 했을 때는 추상적 정보처리가 증가하는 것으로 조사되었다[17].



<그림 1> 계층적 글자[17]

한편, 의학 분야에서도 아동의 구체적 사고와 추상적 사고를 평가하는 방법에 대한 연구들이 진행되어 왔는데, 한 연구[18]에서는 다음과 같이 추상적 사고 과정은 문제 해결을 위해 체계적 접근이 가능하며 고정적이고 일정한 의미로부터 목록화나 분류와 같은 방법으로 일반적인 의미로 개념화시키는 과정으로 정의되고 있다.

Vallacher와 Wegner의 행동 정체성 이론은 사람들마다 추상적 사고 수준에서 차이가 있다고 설명한다[13][16]. 예를 들어, 이를 닦는 경우, 어떤 사람은 단순히 치약을 칫솔에 묻혀 위아래로 닦는 행동으로 생각하지만, 어떤 사람은 충치를 예방하여 건강을 돕기 위한 행동으로 생각한다. 즉, 이를 닦는 것을 칫솔질이라는 구체적인 행동을 먼저 떠올리는 구체적인 사고를 하는 사람이 있는 반면에 건강과 같은 추상적인 행동을 먼저 떠올리는 사람도 있다. 전자보다는 후자가 더 추상적 사고양식을 갖는다고 할 수 있다[13][16].

개인적인 사고 수준에 차이가 있어 같은 행동에 대해서도 그 행동에 대해 어떤 행동의 구체적인 수단에 초점을 두는 “어떻게(how?)”에 초점을 두는 구체적 사고가 강한 사람이 있는가 하면, 상위수준의 목표인 “왜(why?)”에 초점을 두는 추상적 사고가 강한 사람이 있다. 행동 정체성 척도 이론은 개인적 사고 수준의 차이에 따라 행동의 추상적 목표와 구체적 수단에 초점을 두는 것을 설명한다[19].

2.2 컴퓨터 프로그램 이해력의 개념과 특성

본 연구에서는 추상화의 가장 기본적인 개념인 구조적 프로그램의 개념을 기반으로 컴퓨터 프로그램 이해력에 대한 조작적 정의로 다양한 프로그램 중에서 구조적 프로그램의 요소인 순차, 선택, 반복 요소를 기반으로 기 작성이 된 컴퓨터 프로그램을 관독하여 그 결과를 예측하는 능력이라 정의하였다[20][21]. 많은 연구들이 ‘알고리즘 시각화’를 진행하고 있으며 학술 정보를 구하는 국내의 한 사이트(<http://academic.naver.com>)에 따르면, 알고리즘 시각화에 대한 기존의 연구가 915편에 다다르고 있다. 이 중에서 대부분의 경우는 프로그램 작성과 작성된 프로그램의 시각화에 초점을 두고 있다. 한편, ‘컴퓨터 프로그램 이해력’의 경우는 대부분 컴퓨터를 활용한 교육이 다른 영역에서의 이해력이나 능력을 향상시킨다는 연구였다[22][23].

기존 국내의 연구들 중에서 대학에서 프로그래밍 언어 학습을 위해 순서도와 C 언어로 된 소스 코드를 제공하여 프로그래밍 초보자를 위한 학습 시스템을 개발한 사례가 있다[24]. [24]의 연구에서는 대학생들의 논리적 사고력과 문제 분석 기술 등에 초점을 두었으며 실제 프로그램을 구현하기 위해서 순서도를 활용하여 순서도가 진행되는 순서를 시각적으로 제시하며, 실제 시스템에서 C 코드를 제공함으로써 프로그래밍 언어에 대한 이해를 높였다.

3. 연구 방법

3.1 연구대상

본 논문은 연구자 근무지역의 J 인문계 고등학교 1학년 학생들 300명 (남학생 : 149명, 여학생 : 151명)과 D 인문계 여고 3학년 76명을 대상으로 설문조사를 실시하였다. 설문 조사 기간은 2014년 7월과 12월이었다. 학생들이 17주 수업을 모두 이수하고 학기말 시험이 끝난 직후에 설문을 실시하였다. 설문조사의 내용은 설문 대상자들의 추상적 사고 수준과 프로그램에 대한 이해력을 테스트하고 학생들의 학업성취도에 대한 자가진단을

실시하는 것이었다. 설문조사 후에는 학생들에게 5,000원 권의 문화상품권을 주었다.

3.2 추상적 사고 수준과 언어친밀성

본 연구는 Vallacher와 Wegner의 행동 정체성 척도(Behavioral Identification Form: BIF)를 학생들의 추상적 사고 수준을 측정하는데 사용하였다. Vallacher와 Wegner는 원래 행동에 대해서 낮은 단계의 해석과 높은 단계의 해석 중에 하나를 선택하게 하는 25개의 문항을 제안하였다[13]. 추상적 문항을 많이 선택한 응답자는 추상적 사고, 즉 상위해석 수준을 가지고 있다고 분류하며, 구체적 항목을 많이 선택한 응답자는 구체적 사고, 즉 하위해석 수준을 가지고 있다고 분류한다. 행동정체성 척도 개발을 위한 Vallacher와 Wagner의 연구에서 문항들의 내적 일관성은 .85로 신뢰할만하고 항목-전체 상관관계는 .28에서 .48 사이로 조사되었다[13]. 본 연구는 Vallacher와 Wagner가 제시한 방법을 따라 <표 1>과 같이 행동정체성 척도 25개 항목에 대해 구체적인 해석에 대해서는 0점, 추상적인 해석에 대해서는 1점을 부여한 후에 모두 더한 점수를 추상적 사고 수준 값으로 측정하였다[13].

<표 1> 추상적 사고 수준 측정을 위한 설문 항목들[13]

	<상황>	두 가지 대안들
1	목록 만들기	① 조직화하기* ② 할 일들을 적기
2	독서	① 인쇄물의 줄 따라 읽기 ② 지식을 얻기*
3	군대생활 하기	① 국가의 방어를 도와주기* ② 군 입대에 신청하기
4	세탁하기	① 옷에서 냄새 없애기* ② 옷들을 세탁기에 넣기
5	사과 집기	① 무언가 먹을 것을 획득하기* ② 나뭇가지에서 사과를 따기
6	나무 베기	① 도끼를 휘두르기 ② 빨감을 얻기*

		* : 높은 추상적 사고 수준

즉, 각 문항에 대해서 추상적인 해석수준이 서로 다른 두 개의 대안을 제시한 후, 추상성이 높은 답을 몇 개 선택하였는지 개수를 세어 합한 결과를 추상적 사고 수준 값으로 정의하였다. 만

일, 행동 정체성 척도를 이용하여 학생들의 추상적 사고 수준을 측정하고 프로그래밍 이해력을 평가한 후, 이들 간의 관계를 학업성취도와 함께 분석한다면 학생들의 추상적 사고가 프로그래밍 능력에 어떻게 영향을 미치며 어떤 방향으로 교수자가 학생을 지도할 수 있을지 방향성을 제시할 수 있을 것이다.

프로그램 언어친밀성은 학생들에게 설문을 하기 직전까지 배운 프로그램 언어의 이름을 기술하라고 설문하여 개수로 측정하였다. 그 결과 최소 1 개부터 4개의 언어까지 배운 학생이 있었으며, 언어친밀성 단계를 상위 집단과 하위집단으로 구분할 때, 한 개의 언어를 배운 학생 (347명)과 두 개 이상의 언어를 배운 학생 (29명)을 대상으로 나누었다.

3.3 컴퓨터 프로그램 이해력 측정

본 연구에서는 프로그래밍 언어로 C언어와 스크래치를 선택하였다. 단, 모든 학생들에게 두 언어를 지도한 것은 아니며, 남학생 집단은 지역 내에서 스크래치를 지도하는 학교의 섭외가 불가능하여 C언어를 학습한 학생을 대상으로 하였으며, 여학생의 경우는 C언어를 학습한 학교 1곳과 스크래치 언어를 학습한 학교 1곳을 정하여 학생들의 프로그램 이해력을 조사하였다.

기본적인 프로그래밍 요소인 순차, 선택, 반복 문장으로 이루어진 간단한 프로그램을 <표 2>와 같이 제시하였고, 이들은 [25]의 연구를 기반으로 하였다. [25]의 연구는 남학생은 C 언어를, 여학생은 스크래치 언어를 학습한 후, 프로그램 이해에 대한 비교 분석을 실시한 연구였으며, 언어를 지도한 교사가 달라, 성별에 대한 분석 및 C 언어를 학습한 여학생에 대한 분석은 불가능하였다.

본 연구에서 채택한 프로그램은 4 문제였으나, 한 문제는 다음 프로그램과 같이 문장이 너무 단순하여 대부분의 학생이 정답을 맞혔다. 세 가지 프로그램 모두 5~7개 정도의 문장으로 구성되었다. 본 연구에서 선택한 프로그램은 학생들이 배운 내용의 범위를 중심으로 제시하였다.

```
main()
{
    int i, j=3;

    i = j+5;
    j = i*2;

    printf("i=%d, j=%d", i, j);
}
```

<표 2> 프로그램의 종류[25]

프로그램 요소	프로그램
순차	<pre>main() { int total; int apple=2, banana=3; total= apple * 2; total = total + (banana * 3); printf("total = %d", total); }</pre> 
선택	<pre>main() { int no, in=3; if (in + 2 > 5) no = in * 10; else if (in + 3 > 5) no = in * 20; else no = in * 30; printf("no=%d", no); }</pre> 
반복	<pre>main() { int x, y, z, noloop=1; x=1; y= 1; z= 0; while (noloop < 5) { z = z + x + y; x = y; y = z; noloop++; } printf("z=%d", z); }</pre> 

4. 연구 결과

이 장에서는 우선 연구에서 선택한 요인인 추상적 사고 수준과 프로그램 이해력, 언어친밀성간의 상관관계를 분석한 후, 연구대상자의 추상적 사고 수준 값에 따라서 상위단계와 하위단계로 나누어 두 그룹간의 프로그램 이해력 차이를 조사하였다. 또한 프로그램 이해력에 대해 추상적 사고 수준과 언어친밀성 간 상호작용 효과 및 추상적 사고수준과 성별 간 상호작용 효과를 분석하였다. 마지막으로 학습성취도와 프로그램 이해력 간의 관계를 분석하였다.

4.1 프로그램 이해력과 추상적 사고 수준, 언어친밀성 간의 상관관계 분석

이 절에서는 설문항목인 추상적 사고 수준 값, 순차, 조건, 반복의 프로그램 이해력과 프로그래밍 언어친밀성 간의 상관관계를 분석하였다. <표 3>과 같이 프로그래밍 요소별 점수와 학생들의 언어친밀성, 추상적 사고 수준을 살펴보았다.

<표 3> 상관관계 분석

		순차	조건	반복	언어친밀성
추상적 사고	계수	.04	-.03	.05	.01
	p	.50	.57	.33	.82
순차	계수	1	.61**	.23**	.18**
	p		<.001	<.001	<.001
조건	계수	.61**	1	.30**	.20**
	p	<.001		<.001	<.001
반복	계수	.23**	.30**	1	.23**
	p	<.001	<.001		<.001

** : 0.01 유의수준에서 유의함(양측검정).

그 결과 프로그램에 대한 경험이 높을수록 모든 프로그래밍 요소에 대한 이해력이 높은 것으로 나타났다. 추상적 사고 수준 값은 전체 학생을 대상으로 하였을 때, 다른 요인들과 유의한 상관관계가 없었다. 다음 절에서 추상적 사고 수준 값에 따라 집단을 상위집단과 하위집단으로 나누어 보다 구체적인 추상적 사고 수준과 다른 요인간의 관계를 분석하였다.

4.2 추상적 사고 수준별 프로그램 이해력 차이 분석

이 절에서는 추상적 사고 수준 값의 평균을 구한 후, 이를 중심으로 평균보다 낮은 집단(하위 집단)과 평균보다 높은 집단(상위 집단)으로 구분한 후, 집단 간 프로그래밍 이해력에 차이가 있는지를 살펴보았다. 추상적 사고 수준이 높은 집단과 낮은 집단의 구분은 Buchanan 외 2인이 사용한 중앙값 분할(median split) 방식[17]을 사용하였다.

본 연구에서 추상적 사고 수준의 중앙값은 전체 학생에 대해 33명이 응답한 14였다. 본 연구에서는 중앙값보다 낮은 학생들(179명)은 사고 수준 낮은 집단으로, 중앙값과 같거나 높은 학생들(197명)은 사고 수준 높은 집단으로 구분하였다.

분석결과는 다음 <표 4>과 같이 전체 학생의 경우에 추상적 사고 수준별로 프로그램 이해력 간의 평균차이는 발생하지 않음을 알 수 있었다. 다음은 전체 집단을 대상으로 분석하지 않고 남학생과 여학생 집단을 구별하여도 차이가 없는지를 살펴보았다. 즉, 남학생과 여학생을 구분하였을 때, 추상적 사고 수준의 차이가 프로그램 이해력에 차이를 가져오는지를 확인하였다.

<표 4> 추상적 사고 수준별 프로그램 이해력에 대한 t-검증결과

종속변수	추상적 사고 수준	평균(표준편차)		t	df	유의 확률(p)
		하위	상위			
순차	하위	1.18	(.81)	-.68	374	.50
	상위	1.24	(.81)			
조건	하위	.48	(.50)	.41	374	.68
	상위	.46	(.50)			
반복	하위	.06	(.24)	-1.18	372.15	.24
	상위	.10	(.29)			
프로그램 전체	하위	1.73	(1.28)	-.80	374	.61
	상위	1.80	(1.29)			

종속변수	Levene의 등분산 가정	검정 유의 확률		t	df	유의 확률(p)
		F	확률			
순차	등분산이 가정됨	.08	.78	-1.18	374	.24
조건	등분산이 가정됨	.01	.93	-.68	374	.50
반복	등분산이 가정되지 않음	5.29	.02	-.80	374	.61
프로그램 전체	등분산이 가정됨	.01	.92			

우선 남학생의 경우에는 전체 149명 중에서 추상적 사고 수준이 상위인 집단은 91명, 하위인 집단은 58명이었다. <표 5>과 같이 반복문의 경우에 추상적 사고 수준 간에 통계적으로 유의한 차이가 발생함을 알게 되었다.

<표 5> 남학생의 추상적 사고 수준의 차이가 프로그램 이해에 미치는 영향

종속변수		Levene의		t	df	유의 확률 (p)
		등분산	검정유의 확률			
	F					
순차	등분산이 가정됨	1.66	.20	-0.86	147	.40
조건	등분산이 가정됨	1.70	.19	-0.75	147	.45
반복	등분산이 가정되지 않음	17.61	<.001	-2.25	137.98	.03
프로그램 전체	등분산이 가정됨	.07	.80	-1.21	147	.23

즉, 본 연구의 분석 대상 중 남학생의 경우에는 추상적 사고 수준이 높은 학생들이 통계적으로 유의하게 반복문을 담은 프로그램을 더욱 잘 이해함을 알 수 있고, 순차나 선택의 경우에는 유의한 차이가 없음을 확인하였다. 평균에서 드러나듯이 순차 < 조건 < 반복 순으로 학생들이 느끼는 프로그램의 난이도는 점점 높아짐을 확인할 수 있었다. 결국, 어려운 프로그램에 대한 이해에서 추상적 사고력의 차이가 영향을 미침을 알 수 있었다.

여학생의 경우는 전체 227명 중에서 추상적 사고 수준 상위집단 106명, 하위집단 121명에 대한 프로그래밍 이해력을 조사하였다. 그 결과, <표 6>과 같이 추상적 사고 수준 간에 프로그램 이해력의 모든 요소에 대해서 통계적으로 유의한 차이를 드러내지 않았다. 즉, 여학생의 경우에는 추상적 사고 수준의 차이가 프로그램 이해력에 영향을 미치지 못하였다.

<표 6> 여학생의 추상적 사고 수준의 차이가 프로그램 이해에 미치는 영향

종속변수		Levene의		t	df	유의 확률 (p)
		등분산	검정유의 확률			
	F					
순차	등분산이 가정됨	.12	.73	-0.09	225	.93
조건	등분산이 가정됨	.37	.54	1.12	225	.26
반복	등분산이 가정됨	.01	.94	-0.06	225	.95
프로그램 전체	등분산이 가정됨	.45	.50	.37	225	.71

4.3 프로그램 이해력에 대한 추상적 사고 수준과 언어친밀성, 성별 간의 상호작용 효과

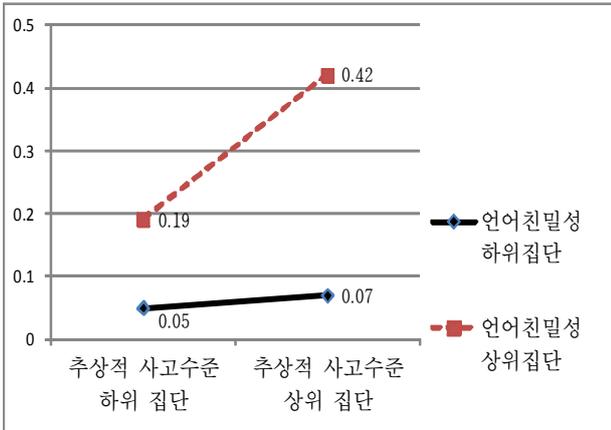
이 절에서는 우선 추상적 사고 수준과 언어친밀성이 학생들의 프로그램 이해력에 어떤 영향을 미치는지 살펴보았다. 이를 위해 상호작용 효과(interaction effect) 분석을 실시하였다. 상호작용 효과는 하나의 독립변수와 하나의 종속변수의 관계만을 살피지 않고, 다른 독립변수를 함께 고려하여 종속변수에 미치는 영향을 함께 살핀다. 즉, 프로그램 이해력에 있어서 추상적 사고 수준과 언어친밀성 간에 상호작용 효과가 있다면, 프로그램 이해력을 높이기 위한 교육에서는 추상적 사고 수준과 언어친밀성을 독립적으로 두지 않고 함께 고려를 해주어야 시너지 효과가 생김을 알 수 있었다.

제시한 프로그램들 중에서 조건이나 순차 문제의 경우에는 상호작용 효과가 나타나지 않았다. 다음 <표 7>에서와 같이 모든 학생들에 대해서 추상적 사고 수준과 언어친밀성 수준의 차이에 따라서 프로그램 중 반복문을 포함한 프로그램에 대한 이해에서 각 주 효과는 통계적으로 유의하지 않았으나, 이 두 요인의 상호작용 효과는 존재하는 것으로 드러났다.

<표 7> 추상적 사고 수준과 언어친밀성 수준 간의 상호작용 효과

변량원	자승합	자유도	평균 자승	F	유의 확률 (p)
추상적 사고 수준 (A)	.44	1	.44	1.44	.44
언어친밀성 수준 (B)	1.56	1	1.56	5.12	.27
A*B	.30	1	.30	4.54	.03

<그림 2>에는 추상적 사고 수준과 언어친밀성 수준을 바탕으로 집단 간에 반복 문제를 해결한 평균의 차이를 보여주고 있다. 추상적 사고 수준이 낮고 언어친밀성 수준도 낮은 집단의 평균은 0.05, 추상적 사고 수준이 낮고 언어친밀성 수준은 높은 집단의 평균은 0.19, 추상적 사고 수준이 높고 언어친밀성 수준은 낮은 집단의 평균은 0.07, 추상적 사고 수준이 높고 언어친밀성 수준도 높은 집단의 평균은 0.42이었다.



<그림 2> 반복문을 포함한 프로그램 이해력에 대한 추상적 사고 수준과 언어친밀성 수준 간의 상호작용

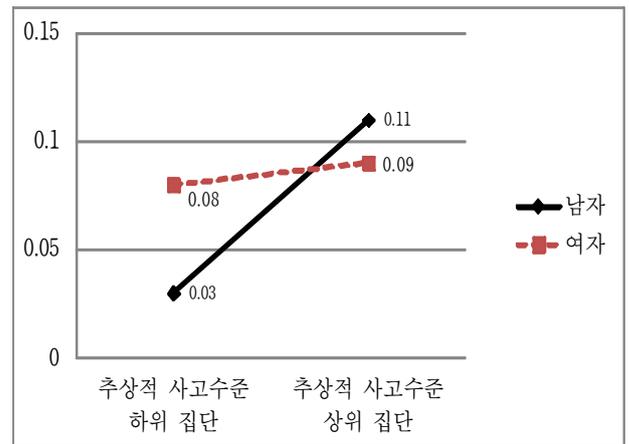
모든 학생들은 추상적 사고 수준이 낮을 때에는 언어친밀성이 높거나 낮을 때 통계적으로 유의한 차이가 나타나지 않았다. 하지만, 추상적 사고 수준이 높을 때에는 언어친밀성이 높아지면 낮을 때에 비해서 통계적으로 유의하게 이해력이 급상승함을 알 수 있었다. 결국 두 독립 요인들 간에는 서로 상호작용 효과가 존재하여 두 가지 요인을 함께 향상시킬 수 있는 교육적 대안이 필요함을 알 수 있다.

다음은 프로그램 이해력에 대한 추상적 사고 수준과 성별 간 상호작용 효과를 분석하였다. <표 8>에서와 같이 반복문을 포함한 프로그램 이해력에 대하여 모든 학생들에 대해 추상적 사고 수준과 성별 간 차이는 순서와 조건을 포함한 프로그램과 동일하게 통계적으로 유의하지 않았다. 상호작용 효과도 존재하지 않았다. 즉, 성별의 차이와 추상적 사고 수준의 차이는 서로 영향을 미치지 않았다.

<표 8> 반복문을 포함한 프로그램 이해력에 대한 추상적 사고 수준과 성별 간의 상호작용 효과

변량원	자승합	자유도	평균 자승	F	유의 확률 (p)
추상적 사고 수준 (A)	.16	1	.16	1.11	.48
성별 (B)	.02	1	.02	.15	.76
A*B	.15	1	.15	2.04	.15

단, 평균의 차이를 보기 위해서 <그림 3>에는 추상적 사고 수준과 성별 간에 반복 문제를 해결한 평균의 차이를 보여주고 있다. 추상적 사고 수준이 낮은 남학생 집단의 평균은 0.026, 추상적 사고 수준이 낮은 여학생 집단의 평균은 0.08, 추상적 사고 수준이 높은 남학생 집단의 평균은 0.11, 추상적 사고 수준이 높은 여학생 집단의 평균은 0.09였다.



<그림 3> 반복문을 포함한 프로그램 이해력에 대한 추상적 사고 수준과 성별 간 프로그램 이해력

결과적으로 본 연구의 집단에서 남학생의 경우에는 추상적 사고 수준에 따른 집단 간의 차이가 매우 큰 반면, 여학생의 경우에는 거의 차이를 보이지 않았다. 또한, 남학생의 경우, 추상적 사고 수준이 낮은 집단은 여학생보다 프로그램 이해력이 낮은 반면, 추상적 사고 수준이 높은 집단은 여학생보다 프로그램 이해력이 높았다. 마지막으로 성별과 언어친밀성 간에도 상관관계는 존재하지 않았다.

4.4 학업성취도 분석

이 절에서는 프로그램 이해력과 학업성취도 간의 관계를 분석하였다. 다음 <표 9>에서와 같이 모든 학생들에 대해서 3 가지 프로그램 이해력과 학업성취도 간의 회귀분석을 실시하였다.

회귀분석결과를 살펴보면, 이들 간에 모두 정 (+)적인 관계를 가짐을 알게 되었으며, 통계적으로도 유의하였다. 독립변인들의 설명력(R²)은 24.6%였다. 상대적 중요도는 표준화 계수(β)를 통

해 확인할 수 있었는데, 조건문을 담은 프로그램이 가장 컸고 반복문을 담은 프로그램, 순차 프로그램 순이었다. 결국, 조건문에 비해 순차는 학업성취도와 관계없이 쉬웠고, 반복문은 어려웠던 것을 파악할 수 있었다.

<표 9> 학업성취도와 프로그래밍 이해력 간의 회귀분석 결과

독립변인	비표준화 계수		표준화 계수	t	유의확률 (p)
	B	표준오차	베타		
(상수)	2.194	.091		24.01	<.001
순차	.194	.079	.139	2.44	.02
조건	.604	.131	.268	4.61	<.001
반복	.988	.199	.235	4.97	<.001

종속변수: 학업성취도, $F(3, 372) = 40.46, p < 0.001$
 $R^2 = .246$.

4.5 연구결과 분석

본 연구를 통해 도출된 결과는 다음과 같다. 우선, 전체 연구대상자에 대해서는 추상적 사고 수준과 프로그램 이해력 간의 상관관계는 존재하지 않았다. 이는 성별 간 추상적 사고 수준과 프로그램 이해력 간에 다소 상반된 경향을 드러내어 상쇄되었기 때문이다. 하지만, 학생들의 추상적 사고수준을 상위와 하위집단으로 구분하고 성별을 나누어 프로그램 이해력과의 관계를 분석한 결과, 남학생의 경우에는 추상적 사고 수준이 높은 학생들이 통계적으로 유의하게 반복문을 담은 어려운 프로그램 이해력이 높음을 알 수 있었다. 여학생의 경우에는 추상적 사고 수준의 차이가 프로그램 이해력에 영향을 미치지 못함을 알 수 있었다.

프로그램 이해력에 대한 추상적 사고수준과 프로그래밍 언어친밀성 간의 상호작용 효과를 분석한 결과, 반복문을 담은 난이도가 높은 프로그램에 대해서 추상적 사고수준이 높을 때 언어친밀성이 높으면 프로그램 이해력이 월등히 높아짐을 알게 되었다. 즉, 두 요인 간에는 상호작용이 존재함을 알게 되었다.

마지막으로 기타 분석에서 프로그램 이해력과 학업성취도 간의 관계를 분석한 결과, 학업성취도

가 높은 학생일수록 3 가지 프로그래밍 요소에 대해서 잘 이해하고 있음을 알 수 있었다. 프로그램 이해력에 대해 성별과 추상적 사고 수준 간에 상호작용 효과가 존재하는지 여부를 조사한 결과가 두 요인에 대해서는 상호작용 효과는 존재하지 않았다.

5. 결론 및 제언

본 연구는 행동 정체성 척도를 이용하여 학생들의 추상적 사고 수준을 측정된 후에 학생들의 프로그래밍 이해력과 어떤 관련성을 갖는지를 살펴보았다. 설문조사에는 인문계 고등학교의 학생들이 참여하였으며, 한 학기 동안 C와 스크래치 프로그래밍 언어 중에 한 언어를 학습한 후, 질문에 응답하였다.

연구 결과는 다음과 같다. 프로그래밍 이해력이 높은 학생들이 학업성취도도 높음을 알게 되었으며, 상대적으로 어려운 프로그램을 이해할수록 학업성취도와 관련성이 높음을 알 수 있다. 언어친밀성과 학업성취도도 정(+)의 상관관계를 갖는 것으로 드러났다. 프로그램의 내용이 단순했음에도 불구하고 순차문만 담은 프로그램은 2점 만점에 평균 1.22, 조건문을 담은 프로그램은 1점 만점에 0.47, 반복문을 담은 프로그램은 1점 만점에 0.08로 각각 78%, 47%, 8%의 학생들만이 정답을 맞힌 것으로 드러났다.

추상적 사고 수준 값은 전체 학생을 대상으로 하였을 때, 다른 요인들과 유의한 상관관계가 없었다. 하지만, 추상적 사고력의 차이는 남학생의 경우 반복문을 해결할 때 드러났으며, 여학생의 경우는 통계적으로 유의하지 않았다. 결국, 전체 학생을 대상으로 분석하였을 때에는 집단 간 효과 차이로 인해 상관관계가 상쇄되어 드러나지 않았다.

또한, 순차와 반복문을 이해하는 데는 유의한 차이가 없었다. 즉, 추상적 사고력은 문제의 난이도가 높아질수록 문제해결에 도움을 주고 있음을 확인하게 되었다. 또한, 추상적 사고 수준과 언어친밀성이 프로그램 이해력에 어떻게 영향을 미치는지 분석한 결과, 각각의 독립변인에 대해서는 통계적으로 유의하지 않았으나, 반복문의 경우 프

로그래밍 이해력에 대하여 추상적 사고 수준과 언어친밀성은 상호작용 효과가 있는 것으로 드러났다. 한편, 추상적 사고 수준과 성별 간의 상호작용 효과가 95% 신뢰구간에서는 유의하지 않았다.

본 연구의 실험으로부터 다음과 같은 결론을 얻었다. 프로그래밍 능력을 높이기 위해서는 적절한 난이도의 문제를 통해 추상적 사고 수준을 높여야 한다. 또한, 추상적 사고와 프로그래밍 언어에 대한 경험인 언어친밀성 간에 상호작용 효과가 존재하기 때문에 추상적 사고력 향상에 도움을 주기 위해서는 다양한 언어에 대한 경험이 추가되어야 한다. 두 요소가 함께 고려되어 컴퓨터 프로그래밍 교육이 진행된다면 교육의 효과는 급증할 것이다.

추상적 사고 수준을 향상시키기 위한 한 가지 대안은 학생들의 심리적 거리를 증가시키는 것이다. 예를 들어, Förster의 연구[26]에서는 시간적 거리가 문제유형에 따라 문제해결력에 차이가 있는지 조사하였다[26]. 그들의 연구결과에서 먼 미래 조건의 실험참가자는 가까운 미래 조건의 실험참가자보다 더 많은 통찰문제를 푸는 것으로 나타났고, 가까운 미래 조건의 실험참가자는 먼 미래 조건의 실험참가자보다 분석 문제를 잘 푸는 것으로 나타났다. 즉, 이 연구가 시사하는 바는 먼 시간관점을 갖게 되면 추상적 사고가 증가하여 통찰력을 향상 시킨다.

또한, 해석수준이론에 의하면 사람들은 어떤 대상에 대해 심리적 거리가 먼 경우에는 상위수준으로 해석하지만, 심리적 거리가 가까운 경우에는 하위수준으로 해석하려는 경향이 있다고 한다[7]. 심리적 거리가 먼 경우에는 상위 목표와 관련 있는 추상적이고 본질적인 가치에 부합되는 상위해석수준을 하게 된다[7][26]. 먼 심리적 거리를 가질 수 있는 교육이 진행된다면 학생들의 추상적 사고 수준도 함께 높일 수 있는 계기를 마련하게 될 것이다.

친밀성에 대한 연구는 두 가지 상반된 주장을 하고 있다. 한 분야에서는 친밀성과 지식이 어떤 영역에 대한 나무 대신 숲을 보게 해주어 추상적 사고력에 도움을 준다고 보고하고 있다[27]. 또한, MacDonald의 연구[28]에서는 친밀성에 해당하는 경험이 항상 추상적 사고력과 긍정적인 관계를

갖게 하는 것은 아니라고 보고하면서도 경험을 통해 터득한 지식을 다른 새로운 업무에 적용하게 되어 추상적 사고력에 도움을 준다고 보고하고 있다.

그러나, Förster의 다른 연구[29]인 새로움과 친밀성에 관한 연구에서 낯선 것에 대한 문제 해결에서 사람들은 추상적 사고수준이 높을수록 문제해결력이 높다는 연구결과를 발표하였다. 그러므로 친밀성이 문제해결에 미치는 영향에 관해서는 향후 보다 많은 연구가 진행되어야 할 것이다. 본 연구에서는 전자의 경우에 해당되었는데, 학생들이 이전에 경험한 언어가 C나 스크래치가 대부분이고 기타 자바나 비주얼 베이직이 있어서 친밀성이 추상적 사고력에 긍정적으로 작용하였다.

본 연구는 두 학교에 대해서만 실험을 실시하였다는 한계점을 갖는다. 본 연구의 결과를 일반화시키기 위해서는 향후 보다 많은 학생들에 대한 분석이 필요하다. 또한, 프로그래밍 언어들은 절차적 또는 객체지향적 등과 같이 다양한 패러다임을 기반으로 하고 있다. 또한 최근의 교육용 프로그래밍 언어들을 스크래치와 같이 대부분 블록기반의 언어들이다. 본 연구에서는 전통적인 언어인 C를 기반으로 하였는데 향후에는 다양한 언어들에 대한 특징도 함께 가미된 분석이 필요하다.

참고 문헌

- [1] 트래블뉴스와이어 (2015). **소프트웨어 중심 사회 위한 인재양성 추진계획**. <http://tnews.wire.com/45028>에서 인용.
- [2] 한국경제 (2013). **인재교육-창의력, 경제발전 핵심**. <http://www.wowtv.co.kr/newscenter/news/view.asp?bcode=T30001000&artid=A201311060364&arttype=V>에서 인용.
- [3] 박권생 (2005). 창의력과 통찰문제 해결능력. **사고개발**, 10, 23-40.
- [4] 조연순, 성진숙, 이해주 (2008). **창의성 교육**, 서울이화여자대학교 출판부.
- [5] 김영정 (2004). 비판적 사고: 비판적 사고와 공학교육. **공학교육**, 11(2), 94-101.

- [6] Glick, J., & Wapner, S. (1968). Development of transitivity: Some findings and problems of analysis. *Child Development*, 621-638.
- [7] Trope, Y., & Liberman, N. (2003). Temporal construal. *Psychological Review*, 110(3), 403-421.
- [8] 김병원 (2003). 과학적 사고 과정에 드러나는 창의성 기능. **대한사고개발학회 학술발표대회 발표논문집**, 81-89.
- [9] Runco, M. A. (1994). *Creative and imaginative thinking*. Encyclopedia of Human Behavior, 2.
- [10] Kramer, J. (2007). Is abstraction the key to computing? *Communications of the ACM*, 50(4), 37-41.
- [11] Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- [12] Wiedenbeck, S. (2005). Factors affecting the success of non-majors in learning to program. *Proceedings of the ACM first international workshop on Computing education research*, 13-24.
- [13] Vallacher, R. R., & Wegner, D. M. (1987). What do people think they're doing? Action identification and human behavior. *Psychological Review*, 94(1), 3-15.
- [14] Burgoon, E. M., Henderson, M. D., & Markman, A. B. (2013). There are many ways to see the forest for the trees: A tour guide for abstraction. *Perspectives on Psychological Science*, 8(5), 501-520.
- [15] Navon, D. (1977). Forest before trees: The precedence of global features in visual perception. *Cognitive Psychology*, 9, 353-383.
- [16] Vallacher, R. R., & Wegner, D. M. (1989). Levels of Personal Agency : Individual variation in action identification. *Journal of Personality and Social Psychology*, 57(4), 660-671.
- [17] Buchanan, T. W. & Lovallo, W. R. (2001). Enhanced memory for emotional material following stress-level cortisol treatment in humans. *Psychoneuroendocrinology*, 26(3), 307-317.
- [18] 김인주, 신민섭 (1999). 인물화 검사와 KEDI-WISC 공통성 소검사를 통한 소아 정신과 아동의 구체적 사고와 추상적 사고의 평가. **소아·청소년정신의학**, 10(2), 186-194.
- [19] Kim, H. & John, D. R. (2008). Consumer response to brand extensions: Construal level as a moderator of the importance of perceived fit. *Journal of Consumer Psychology*, 18(2), 116-126.
- [20] Hausler, P. A., Pleszkoch, M. G., Linger, R. C., & Hevner, A. R. (1990). Using function abstraction to understand program behavior. *IEEE Software*, 7(1), 55-63.
- [21] Shaw, M. (1984). Abstraction techniques in modern programming languages. *IEEE Software*, 4, 10-26.
- [22] Storey, M. A., Wong, K., & Müller, H. (1997). How do program understanding tools affect how programmers understand programs? *Proceedings of the 4th IEEE Working Conference on Reverse Engineering*, 12-21.
- [23] Aldrich, J., Kostadinov, V., & Chambers, C. (2002). Alias annotations for program understanding. *ACM SIGPLAN Notices*, 37(11), 311-330.
- [24] 박경욱, 오경숙, 류남훈, 이혜미, 김응곤 (2010). 기초 알고리즘을 활용한 프로그래밍 언어 학습 시스템. **한국전자통신학회 논문지**, 5(1), 66-73.
- [25] Park, C. J., & Hyun, J. S. (2014). Effects of abstract thinking and familiarity with programming languages on computer programming ability in high schools, *Proceedings of the IEEE Conference on Teaching, Assessment, and Learning*, 468-473.
- [26] Förster, J., Friedman, R. S., & Liberman, N. (2004). Temporal construal effects on abstract and concrete thinking: consequences for insight and creative

cognition. *Journal of personality and social psychology*, 87(2), 177-189.

- [27] Alba, J. W., Hutchinson, J. W. (1987). Dimensions of consumer expertise. *Journal of consumer research*, 13(4), 411-454.
- [28] MacDonald, L., Holdsworth, P. (1997). *Perfect Practice: The Coaching Edge*. Centax Books & Distribution, Canada.
- [29] Förster, J. (2009). Cognitive consequences of novelty and familiarity : How mere exposure influences level of construal. *Journal of Experimental Social Psychology*, 45(2), 444-447.



박찬정

1988 서강대학교
전자계산학과(공학사)
1990 한국과학기술원
전산학과(공학석사)

1998 서강대학교 대학원 전자계산학과(공학박사)
1990~1994 한국통신 소프트웨어연구소 전임연구원
1998~1999 한국통신 멀티미디어연구소 전임연구원
1999~현재 제주대학교 컴퓨터교육과 교수, 교육과학연구소 연구원
2013~2015 제주대학교 교육과학연구소 소장
관심분야: 시간관, 추상적 사고력, 문제해결력, 창의인성교육, 정보문화

E-Mail: cjpark@jejunu.ac.kr



현정석

1991 서강대학교
경영학과(경영학사)
1993 서강대학교 대학원
경영학과 (경영학석사)

1998 서강대학교 대학원 경영학과(경영학박사)
2002~현재 제주대학교 경영정보학과 교수
2007 제주대학교 연구업적 우수교수상 수상
2008 제주대학교 대학을 빛낸 교수상 수상
2012 특허청장상 수상
관심분야: 마케팅, 행동의사결정론, 트리즈, 창의성 교육, 영재교육

E-Mail: jshyun@jejunu.ac.kr

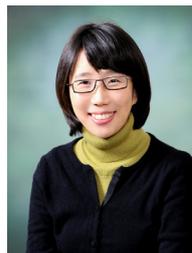
진희란



2003 제주대학교
컴퓨터교육과(학사)
2007 제주대학교
컴퓨터교육과(석사)

2004~2006 제주중앙여자고등학교 교사
2007~2008 한국뷰티고등학교 교사
2009 성산고등학교 교사
2010~2012 한국뷰티고등학교 교사
2013~현재 제주대학교 사범대학부설고등학교 교사

관심분야: 컴퓨터교육, u-러닝
E-Mail: heuilan@naver.com



정혜선

2000 제주대학교
컴퓨터교육과(학사)
2002~현재 제주특별자치도교육청
중등교사

관심분야: 컴퓨터교육, 융합과학
E-Mail: sunny0506@korea.kr