수학 알고리즘 기반의 코딩 교육

장희선
평택대학교 교수

I. 서론

인공지능, 사물인터넷, 가상·증강현실, 로봇, 자율주행차 등 4차 산업혁명 시대의 제품 및 서비스 개발의 핵심은 소프트웨어이다[3],[6],[7],[9]. 이에 따라 선진국에서는 소프트웨어 인력 양성을 위한 조기교육을 실시하고 있다. 미국은 연방정부 차원에서 교육과정 표준을 마련하여 컴퓨터과학적 사고, 문제해결을 위한 의사소통과 협력, 프로그래밍, 컴퓨터와 통신, 정보생활의 5개 영역으로 나누어 교육한다[4],[9]. 영국은 학생들이 다른 사람이 만든 프로그램을 사용하는 것보다 만드는 방법을 배울 수 있도록 5~14세의 모든 학생들에게 컴퓨터 프로그래밍 교육을 실시하고 기존 ICT과목을 컴퓨팅 과목으로 변경하여 초중고 필수과목으로 운영하며, 이 외에 에스토니아, 이스라엘, 일본, 인도 등도 국가 주도로 체계적인 소프트웨어 교육과정을 마련하여 조기교육을 시행하고 있다[2],[5],[9].

우리나라도 2018년부터 단계적으로 초중고 교과과정에서 코딩 교육을 시작하였다[4],[13]. 그러나 교육현장에서는 따라하기 식의 블록코딩(Block Coding)[10],[12],[18]과 피지컬 컴퓨팅[2],[4] 등 흥미위주의 교육이 주를 이루고 있어 논리적 사고력과 창의적인 문제해결 능력을 갖춘 수학 알고리즘 기반의 코딩교육에 대한 필요성이 대두되고 있다[4],[8],[13],[19]. 본 고에서는 미국 공군사관학교(US Air Force Academy) 등 세계적으로 코딩 및 프로그래밍 기초 교육을 위해 활용[11],[19],[20],[23]되는 Raptor 순서도 플랫폼[16]을 소개하고 엔트리 블록코딩과 함께 간단한 예제를 설명한다. 그리고 관련자들(645명)의 설문을 통해 과정별 교육도구(프로그래밍 언어), Raptor의 유용성, 교육 실시시기 및 주안점 등의 조사결과를 제시하며, 코딩을 인지하고 있는 경력자(코딩 교육·개발자)와 비경력자 사이의 카이제곱 검정[1]을 이용한 유의성 분석 결과를 설명한다.

---

* 본 내용은 평택대학교 장희선 교수(☎031-659-8283)에게 문의하시기 바랍니다.
** 본 내용은 필자의 주관적인 의견이며 IITP의 공식적인 입장이 아님을 밝힙니다.
*** 본 고는 2017년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No.2017R1E1A1A03070134)

본 연구의 주목적은 지금까지 이루어지고 있는 블록코딩과 피지컬 컴퓨팅 등의 교육 방법을 개선하여 수학 알고리즘 기반의 코딩 교육모델을 새로 정립해야 할 당위성을 제안하는 것이다.

II. 블록코딩 및 Raptor 순서도

일반적으로 컴퓨터를 이용한 문제해결은 ① 문제의 정의, ② 알고리즘 설계, ③ 코딩, ④ 디바이스(스마트폰, PC, 로봇 등) 구현, ⑤ 상품·서비스 제공의 절차로 이루어진다[9],[19]. 미국 공군사관학교의 프로그래밍 교재[19]에서는 ② 알고리즘 설계와 ③ 코딩을 [표 1]의 4단계로 제시하고 있다. 먼저, 달성 목적을 이해하고, 문제해결 절차(작업, 프로세스)를 구조적 프로그래밍(Structured Programming)[9]의 세 가지 논리(순서, 선택, 반복)로 설계하며, 프로그래밍 언어로 구현 후 테스트한다. 특히, 구현 단계에서는 이해가 쉽고 ISO 국제표준 기호[19]를 따르는 Raptor 순서도를 활용하여 수학 알고리즘 기반의 코딩 과정을 설명하고 있다. 교재[19]의 예제(하이-로 숫자 맞추기 게임)를 이용하여 블록코딩과 Raptor 순서도 구현 사례를 설명하며, 초중고 수학 교육과정 기반의 다른 예들은 CoSuDa 사이트[11]를 참고하길 바란다.

[표 1] 알고리즘 설계 및 코딩 절차

알고리즘 설계 및 코딩	주요 내용
1. Understand the goal to be accomplished (달성 목적 이해)	- 주어진 문제와 문제의 요구사항 이해 - 문제에 대한 의문점 질의 - 문제해결 절차를 스토리보드로 작성 - 문제해결 절차(요구사항) 분석 - 절차들 사이의 상호 충돌 여부 검토
2. Design your solution(문제해결 절차 설계)	- 목적 달성을 위한 작업(프로세스) 나열 - 복잡한 작업은 세부 작업으로 분리 - 순서, 선택, 반복의 논리로 작업 처리
3. Implement your solution(구현)	- 각 작업을 프로그래밍 언어(Raptor 순서도)로 구현 - 세부작업별 테스트(전체 테스트 전)
4. Test your solution(테스트)	- "발생 가능한 모든 경우를 처리하는가" 검토 - 다양한 입력값에 대한 프로그램 수행 - 환류(구현→설계→문제의 이해 검토)

〈자료〉 Steve Hadfield, Troy Weingart, Wayne Brown, An Introduction to Programming and Algorithmic Reasoning using Raptor, United States Air Force Academy, CreateSpace an Amazon.com Company, 2018.

1. 엔트리 블록코딩

블록코딩은 어렵게 느껴지는 텍스트 코딩과 비교하여 흥미를 가지고 소프트웨어를 접할 수 있는 도구로서 스크래치[18], 엔트리[12], Code.org[10] 사이트를 이용하며, 특히 엔트리[12]는 블록코딩과 함께 파이선 소스코드를 자동으로 만들어 준다. [그림 1]은 하이-로(Hi-Low) 숫자 맞추기 게임을 블록코딩과 파이선 코드로 구현한 예이다.

[블록코딩(엔트리)]

```python
# 엔트리봇 오브젝트의 파이선 코드

import Entry

Guess = "0"
Goal = "0"

def when_start():
    Entry.hide_variable("Goal")
    Goal = random.randint(1, 45)
    while not (Guess == Goal):
        Entry.input("1부터 45까지의 수는 무엇일까요?")
        Guess = Entry.answer()
        if (Guess > Goal):
            Entry.print_for_sec("예측값이 너무 큽니다. ", 2)
        else:
            Entry.print_for_sec("예측값이 너무 작습니다. ", 2)
    Entry.print_for_sec("축하합니다. 숫자를 맞추었습니다.", 3)
    Entry.show_variable("Goal")
    Entry.show_variable("Guess")
```

[파이선 코드]

[그림 1] 엔트리 코딩(하이-로 게임)

게임은 1부터 45까지의 로또 번호 중 하나를 변수(Goal)에 저장하고 이 숫자를 맞추는 것으로 사용자가 예측한 값(Guess)이 Goal보다 크면 “예측값이 너무 큽니다”를, 작으면 “예측값이 너무 작습니다”를 출력하며, 숫자를 맞추면 프로그램이 종료된다.

2. Raptor 순서도

공군사관학교 재직 시절, Martin Carlisle(현재는 카네기멜론 대학 근무) 교수 등이 처음으로 개발하였으며, 중국의 칭화대학[23]을 포함하여 전 세계 코딩 및 프로그래밍 기초 교육용 플랫폼으로 활용[14],[15],[20]되는 Raptor는 인터넷에서 무료로 다운로드 할 수 있고, 다음과 같은 장점이 있다[11],[19],[20].

① 세계 표준인 ISO-97N90 기반의 순서도(flowchart) 기호 사용

② 시각적(Visual)인 알고리즘 개발이 가능

③ [순서도구현]-[수행]-[결과확인]으로 개발된 알고리즘의 타당성을 바로 검증

④ 소프트웨어 개발을 위한 사용자-개발자들 사이의 커뮤니케이션 도구로 활용

⑤ C++/C#, 자바, VBA의 소스 코드 자동 생성으로 별도 텍스트 코딩이 불필요

⑥ 구조적 프로그래밍과 함께 객체지향적 프로그래밍 설계 방법 제공

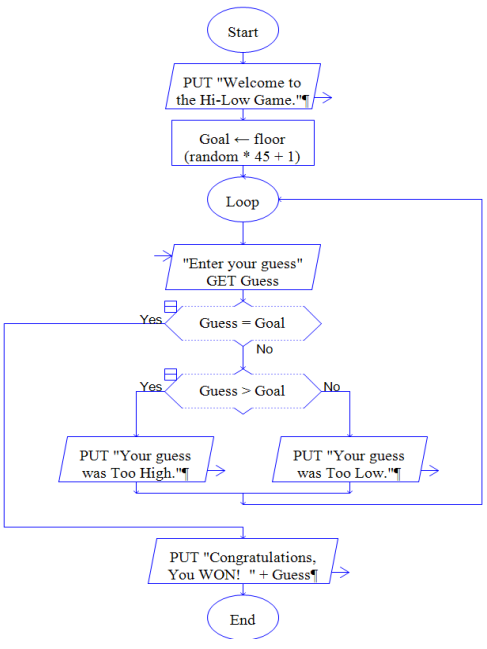
⑦ 프로시저(함수, 모듈, 부프로그램 등) 기능 제공으로 모듈별 공동개발 가능

⑧ 프로그래밍 언어의 문법을 몰라도 수학 알고리즘 학습 가능

⑨ 그래픽, 게임 개발 기능 제공

⑩ 지속적인 업데이트와 함께 매뉴얼 무료 제공

하이-로 게임에 대한 순서도와 자동 생성된 C++ 코드는 [그림 2]와 같으며, Raptor 사용법과 순서도 작성과정은 문헌[16],[19]과 CoSuDa 사이트[11]에 자세히 설명되어 있다.



[순서도]

```
#include <iostream>
#include <string>

using namespace std;
int main()
{
   string raptor_prompt_variable_zzyz;
   int goal;
   int guess;

   cout << "Welcome to the Hi-Low Game." << endl;   goal =floor(random_generator.NextDouble()*45+1);
   while (1)
   {
      raptor_prompt_variable_zzyz ="Enter your guess";
      cout << raptor_prompt_variable_zzyz << endl;
      cin >> guess;
      if (guess==goal)) break;
      if (guess>goal)
      {
         cout << "Your guess was Too High." << endl;      }
      else
      {
         cout << "Your guess was Too Low." << endl;      }
   }
   cout << "Congratulations, You WON!  " << guess << endl;
   return 0;
}
```

[C++ 코드]

〈자료〉 Steve Hadfield, Troy Weingart, Wayne Brown, An Introduction to Programming and Algorithmic Reasoning using Raptor, United States Air Force Academy, CreateSpace an Amazon.com Company, 2018.
C++ 코드는 Raptor 메뉴를 이용하여 제작

[그림 2] Raptor 순서도(하이-로 게임)

블록코딩과 다르게 Raptor는 순서, 선택, 반복의 세 가지 논리만을 이용한 수학적 알고리즘 기반의 코딩 교육에 유리하다. 하지만, 코딩을 처음 접하는 학생들에게 순서도가 다소 어렵다는 의견이 있을 것으로 보인다. 이를 위해 교육과정별로 적절한 코딩 교육도구(프로그래밍 언어), Raptor의 유용성, 교육의 실시시기 및 주안점 등에 대한 의견을 조사하였다.

III. 설문 개요

바람직한 코딩교육의 방향을 설정하기 위해 설문조사를 수행(2017년 11월~2018년 4월)하였으며, [표 2]는 조사내용과 응답자들의 일반사항이다. 2015년 개정된 수학 교육과정과 연계한 교육콘텐츠를 개발하기 위해 코딩교육에 포함되어야 할 수학 교육과정과 함께 교육도구, Raptor의 유용성, 가상현실 콘텐츠의 필요성, 실시시기 및 주안점, 비전공자들에 대한 필요성 등을 조사하였다. 여기서는 지면 관계상 교육도구, Raptor 유용성, 실시시기 및 주안점의 결과만을 설명한다.

[표 2] 설문내용 및 응답자 일반사항

구분			주요 내용
설문 내용	코딩 교육과정	초중고 수학교육과정 정보과학대학	- [수학교육과정] 수와 연산, 도형, 측정, 규칙성, 함수, 기하, 확률 통계, 미적분, 행렬 - [정보과학] 프로그래밍기본, 흐름제어, 모듈화, 자료구조, 정렬검색, 알고리즘, 시뮬레이션, 피지컬 컴퓨팅 - [실생활문제] 경제지표, 이자, 함수와 경제, 평균편차, 로또번호, 석차, 조건검색, 행렬배열, 그래프
	교육도구 (언어) 및 기타	과정별 언어	스크래치(엔트리), 아두이노(라즈베리파이), 앱인벤터 비주얼베이직, C/C++, 자바, 파이선, 자바스크립트
		기타 질문	비주얼도구, Raptor 순서도, 가상현실 콘텐츠의 유용성 코딩교육의 주안점, 교육 실시시기 인문사회학 등 타 전공자들에 대한 교육의 필요성 우리나라 4차 산업혁명 성장동력 기술·서비스
응답자	성별	남 68%, 여 32%	
	직업	학생(초중고) 38%, 대학생 34%, 선생님(초중고) 15%, 연구원 8%, 소프트웨어개발 및 프리랜서 3%, 기타(교수, 강사 등) 2%	
	코딩 경력	비경력자 42%, 경력자 58%	

유효 응답자 수는 645명(남성 68%, 여성 32%)이며, 초중고 38%, 대학생 34%, 선생님 15%, 연구원 8%이고, 코딩교육을 받아 보았거나 개발 경력이 전혀 없는 사람(비경력자)이 42%, 1년 이상 경력자가 58%이다. 설문문항에 대한 신뢰성 분석결과, 크론바하 알파 계수(Chronbach's

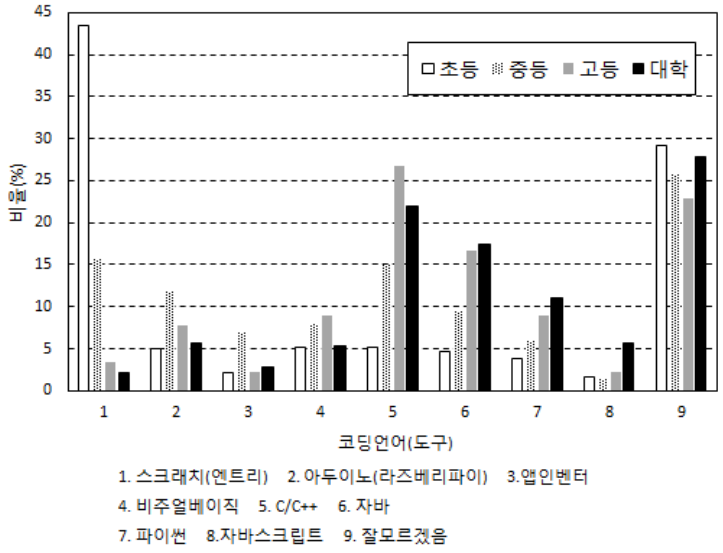
alpha)의 값이 0.743으로 신뢰도가 높다(0.5이상이면 높음[1])고 할 수 있으며, 신뢰수준 99%, 허용오차는 5% 범위에서 조사되었다.

IV. 설문 분석

SPSS 통계 패키지[1]를 이용하며, 먼저, 교육도구, Raptor와 같은 비주얼도구의 유용성, 교육 실시시기 및 주안점에 대한 빈도분석 결과를 설명하고, 코딩 인지도가 높지 않은 비경력자와 1년 이상의 경력자들 사이에 의견의 차이가 있는지에 대한 검정결과를 제시한다.

1. 빈도 분석

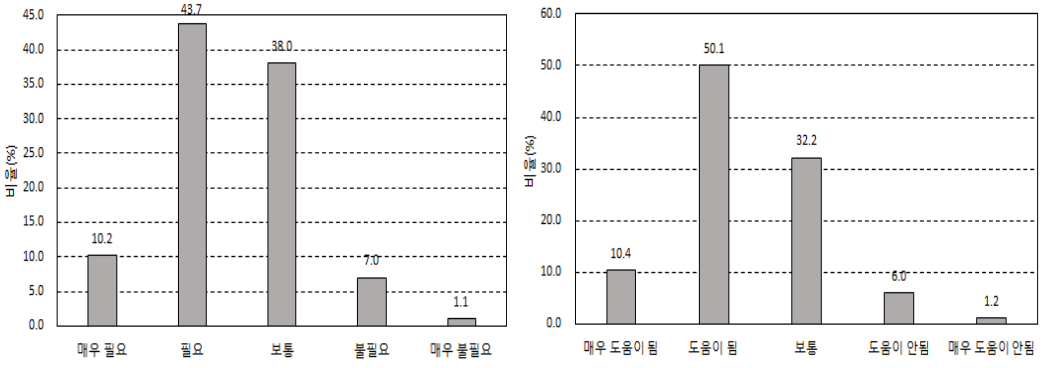
[그림 3]은 코딩교육을 위해 가장 바람직한 도구가 무엇인지에 대한 결과(비율)이다. 초등학교에서는 스크래치(엔트리) 블록코딩이 43.4%로 가장 높고, 중학교는 스크래치(엔트리) 15.7%, C/C++ 15%, 아두이노(라즈베리파이) 11.8% 순이다. 그러나 고등학교 과정은 이와 다르게 C/C++ 26.8%, 자바 16.6%로 실무에서 사용되는 프로그래밍 언어를 이용한 교육을 선호하며, 대학에서도 C/C++ 22%, 자바 17.5%로 비슷하다. 한편, 고등학교에서 파이선에 대한 선호도가 9%, 대학에서는 11%로 나타나 C/C++, 자바와 함께 파이선을 이용한 코딩교육의 필요성도 나타나고 있다.



[그림 3] 바람직한 코딩 교육도구(언어)

이처럼 블록코딩을 제외하면, 바람직한 교육용 언어로 C/C++, 자바, 파이선을 꼽았고, 이는 네덜란드 TIOBE사가 2018년 11월, 발표한 세계 프로그래밍 언어 사용 점유율의 순위(C/C++ 22.7%, 자바 16.8%, 파이선 7.7%)와 일치하는 결과[11],[17],[21],[22]로 응답자 대부분은 실무에서 쓰이는 언어 교육의 필요성을 느끼고 있는 것으로 파악된다.

다음으로 프로그래밍 언어를 접하기 전에 UML(Unified Modeling Language), RSA(Rational Software Architecture, IBM), MDD(Model Driven Development, LG), 그리고 Raptor와 같은 비주얼 도구를 이용한 교육의 유용성(또는 필요성)에 대한 견해는 [그림 4]와 같다.



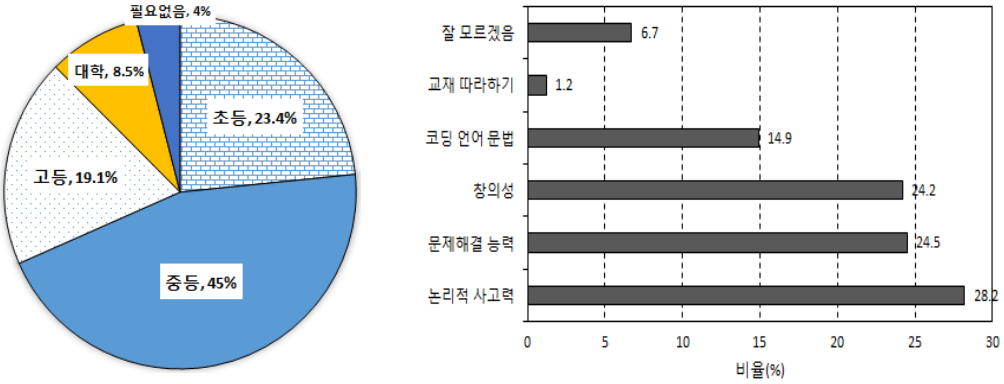
[UML, RSA, MDD] [Raptor 순서도]

[그림 4] 비주얼 도구의 필요성

동일하게 645명을 대상으로 조사하였으며, 이들은 비주얼 도구에 대해 대부분 유용성을 공감하며, Raptor 순서도가 매우 도움이 된다 10.4%, 도움이 된다 50.1%로 긍정적 의견이 60.5%이고, 보통 이상이 92.7%로 도움이 안된다는 견해(7.2%) 보다 긍정적 평가가 많다. IBM의 RSA와 LG MDD와 같은 플랫폼들은 실무에서 활용되고 있어 고등학교 혹은 대학에서 교육용으로 이용할 수 있는 방안이 있으면 좋겠으나, 이는 고가의 소프트웨어로서 활용의 한계가 있다. 그리고 프로그래밍 언어를 접하기 전에 기본적인 수학 알고리즘 기반의 코딩 교육용으로는 너무 복잡한 인터페이스와 기능들로 되어 있어 또 하나의 새로운 도구를 배우는 것과 맞먹는 시간과 노력이 요구된다. 반면, Raptor는 누구나 무료로 인터넷에서 다운로드하여 사용할 수 있고 기본적인 ISO 기반의 6가지 기호(Symbols) 사용법만을 터득하면, 쉽게 배울 수 있어 다른 도구에 비해 교육용으로 매우 적합한 플랫폼이다. 아울러 주요 대학에서도 프로그래밍 언어를 접하기 전에 수학 알고리즘 기반의 추론과 컴퓨팅적 사고력(Computational Thinking) 향상에 도움을 준다고 판단하여 Raptor를 이용한 프로그래밍 기초 교육을 운영하고 있다[19],[23]. 그리고 다른 문헌[14],[15],[20]에서도 프로그래밍

언어의 개념과 함께 Raptor 순서도를 이용한 문제해결과 알고리즘 설계 방법을 소개하고 있다. 따라서 블록코딩, C/C++, 자바, 파이선 등과 함께 Raptor 순서도를 병행하여 교육시킨다면, 논리적 사고력과 창의적 문제해결 능력 향상에 도움이 되는 바람직한 코딩교육이 될 것이다.

[그림 5]는 코딩을 처음으로 교육시켜야 하는 시기와 교육에서 가장 중요하다고 생각하는 요인(주안점)에 대한 결과이다. 대부분 초등학교(23.4%) 보다는 중학교(45%)에서 시작하는 것이 바람직하다는 의견이 많다. 이는 너무 일찍 어려운 코딩을 배우는 경우 소위 "수포자(수학 포기자)"와 비슷한 "코포자(코딩 포기자)"를 길러낼 가능성이 있다고 판단한 것으로 보이며, 자유롭게 기술하게 한 기타 의견에서도 비슷한 우려를 보였다. 이러한 우려로 인해 스크래치(엔트리)와 같은 블록코딩으로 초등학생들에게 흥미를 심어주는 교육 방법을 제시한 의견이 많았던 것으로 판단된다.



[교육 실시 시기] [교육의 주안점]

[그림 5] 코딩교육의 시기 및 주안점

645명의 응답자들에 대해 코딩교육에서 가장 중요하다고 생각하는 것에 대한 조사 결과, 논리적 사고력이 28.2%로 다소 높고, 다음으로 문제해결 능력(24.5%), 창의성(24.2%)이 비슷하며, 코딩 언어 문법이 14.9%이다. 모든 항목이 중요하겠지만, 그 중에서도 논리적 사고력이 문제해결 능력, 창의성 그리고 언어의 문법 보다 높게 나온 것은 수학 알고리즘적 추론(Algorithmic Reasoning)을 기반으로 한 코딩교육의 중요성을 모두 인식하고 있는 것으로 보인다.

2. 유의성 분석

빈도분석 결과가 코딩 경력자(교육·개발 경험, 또는 인지자)와 비경력자들 사이에 차이가 있는가를 유의성 검정을 통해 설명한다. 즉, 귀무가설(H_0: 비경력자와 경력자 사이의 의견은 같다) 하에서

[표 3] 카이제곱 검정(유의확률)

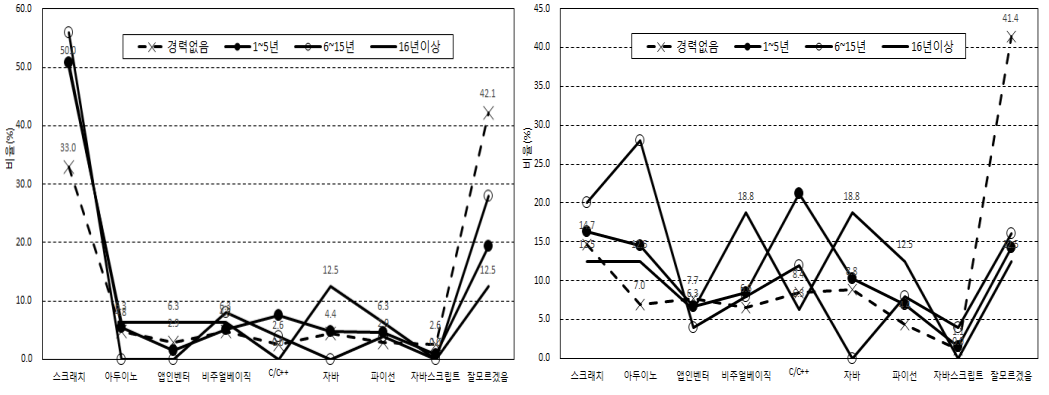
조사 항목	카이제곱 유의확률 (p)*			
코딩 교육도구	초등	0.000	중등	0.000
	고등	0.000	대학	0.000
비주얼 도구 시기 및 주안점	UML,RSA,MDD	0.269	Raptor	0.000
	실시 시기	0.000	주안점	0.053

*유의확률(ρ)〈α=0.05(유의수준)이면 귀무가설 기각(의견에 차이가 있다)

의 카이제곱(Chi-square, x^2) 검정[1] 결과, 유의확률 값은 [표 3]과 같다.

검정 방법은 [표 3]에서 구한 유의확률 값(ρ)이 유의수준(α=0.05)보다 작으면(ρ〈α), 귀무가설을 기각하며, 이 경우 경력자와 비경력자 사이에 의견의 차이가 있다고 본다. 반대로, $\rho \geq \alpha$이면, 귀무가설을 채택하여 비교 그룹 사이의 의견은 같다고 판단한다. 따라서 프로그래밍 언어를 접하기 전 UML, RSA, MDD와 같은 비주얼 도구를 이용한 교육에는 의견의 차이가 없으며, 응답자 중 92% 정도가 보통 이상의 답변으로 필요성에 긍정적이다. 그리고 교육에서 가장 중요하다고 생각되는 요인도 경력과 무관하게 논리적 사고력을 최우선으로 꼽았고, 문제해결 능력, 창의성, 코딩 언어 문법의 순으로 선택하였다.

반면, 경력자와 비경력자 사이의 유의한 결과는 교육도구, Raptor의 유용성 및 교육 실시시기에 나타났고, 교육도구의 의견은 [그림 6, 7]과 같다. 비경력, 경력이 1~5년, 6~15년 그리고 16년 이상의 응답자로 구분하여 비율(%)을 나타내었으며, 비경력자와 16년 이상 경력자의 비율을 그림에 표기하였다. 초등학교에서는 스크래치를 이용한 교육이 바람직하다는 의견이 지배적이고, 이는 경력

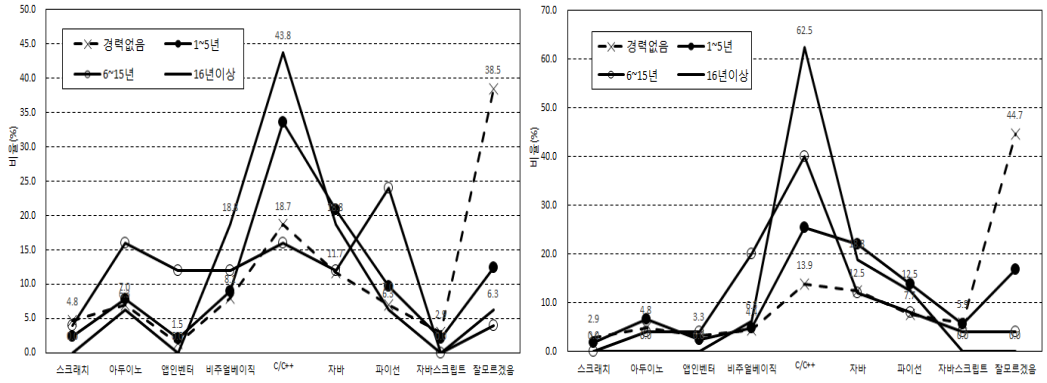


[초등학교] [중학교]

[그림 6] 교육도구의 의견(초등 및 중등)

자들 대부분(50% 이상) 동의하지만, 비경력자들은 33%(잘 모르겠음은 42.1%) 정도가 이를 선호한다. 당연하지만, 비경력자들이 “잘 모르겠음”을 선택한 비율이 가장 높다.

중학교에서도 가장 높은 선호도를 보인 블록코딩(스크래치)에 대해서는 비경력자와 경력자 사이에 큰 차이(13~20%)가 없다. 그러나 아두이노와 C/C++ 언어를 선택한 사람들은 1~5년, 6~15년 경력자가 대부분이며, 16년 이상 경력자 중에는 자바와 비주얼베이직을 중학교 과정에서 교육시켜야 한다(각각 18.8%)고 생각하는 사람이 많았다. 비경력자들은 초중학교에서 스크래치를 적절한 도구로 생각하고 있으나, 특이한 점은 초등학교와 달리, 중학교에서 블록코딩을 선택한 비율이 감소(33%에서 15%로)하고, 대신 C/C++과 자바 선택 비율이 높았다는 점이다.



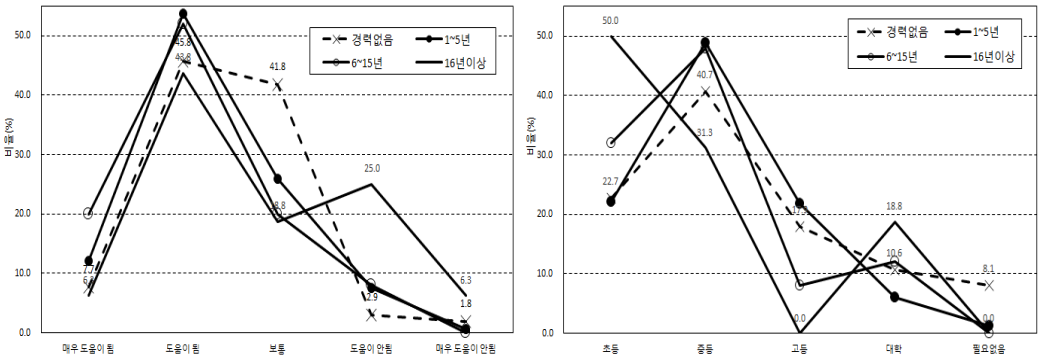
[고등학교] [대학교]

[그림 7] 교육도구의 의견(고등 및 대학)

[그림 7]은 고등학교와 대학의 결과이며, 유의한 차이가 있지만 대부분의 경력자들은 고등학교와 대학에서 가장 적절한 언어로 C/C++를 선택하였다. 대학에서는 경력자들의 25~63%가 C/C++, 두 번째로 자바(12~22%), 세 번째로 파이선(8~14%)을 선호하였다.

고등학교를 보면, 전체적으로 C/C++과 자바를 우선 선택하고, 대학과 달리 3순위로 파이선과 비주얼베이직의 선택 비율이 비슷하며, 비경력자들도 고등학교와 대학에서 C/C++과 자바를 선호하였다. 따라서 고등학교와 대학에서는 블록코딩과 피지컬 컴퓨팅보다는 실무에서 사용되는 프로그래밍 언어를 배우는 것이 바람직하다. 결론적으로, 경력자와 비경력자들 사이에 차이는 있으나, 코딩을 배우기 위한 바람직한 도구(언어)는 초중학교에서 스크래치(블록코딩), 고등·대학에서는 C/C++임을 알 수 있다.

[그림 8]은 Raptor의 유용성과 교육시기에 대한 의견이다. Raptor가 도움이 된다고 생각하는 비율이 가장 높고, 그들 중 1~5년 사이, 처음 코딩을 접하는 사람들이 가장 긍정적인 평가를 하고



[Raptor의 유용성] [코딩교육 실시 시기]

[그림 8] Raptor의 유용성 및 교육 시기의 의견

있어(매우 도움됨은 6~15년 경력자가 많음), 초보자들이 어려운 프로그래밍 언어를 접하기 전에 순서도를 통한 알고리즘 설계의 중요성을 인식하고 있는 것으로 보인다. 반면, 16년 이상 경력자들 중 보통 이상이 68.9%로 긍정적 의견이 많기는 하지만, '도움 안됨'이 25%로 다른 그룹과 비교하여 부정적 의견이 다소 많으며, 비경력자들은 도움이 된다 54%, 보통 이상 95%로 나타났다. 결국, 16년 이상의 오랜 경력자들은 대형 프로젝트 개발을 주로 하고 텍스트 기반의 코딩에 익숙하여 순서도 기반의 알고리즘 설계 방법에 대해 다소 부정적 견해를 가지고 있으나, 대부분 Raptor를 활용한 코딩 교육의 유용성에는 동의하며, 특히 수학 알고리즘 기반의 문제해결(또는 소규모 프로젝트) 절차를 이해하고 설명할 때 좋은 도구로 평가하고 있는 것으로 판단된다.

마지막으로, 교육 시기는 대부분 중학교부터 시작해야 한다는 의견이 지배적이지만, 16년 이상 경력자들의 50%가 초등학교 조기 교육을 원하였다. 또한, 코딩 경험이 없는 사람들도 대부분 초등학교(23%)보다 중학교(41%)를 선택하고 있어, 중학교부터의 교육이 바람직하다고 할 수 있다. 그리고 경력자들 대부분은 코딩교육의 필요성에 동감하고 있으나, 비경력자들의 8% 정도가 필요성에 의문을 제기하였다.

V. 결론

세계적으로 4차 산업혁명의 기술·서비스 개발을 위한 소프트웨어 인력 양성을 중요한 과제로 인식하여 코딩(소프트웨어) 조기교육을 시행하고 있으며, 우리나라도 초중고 교과과정에 의무교육이 실시되어 2019년부터 초등학교에서 17시간(중학교 34시간, 고등학교 일반선택 과목)이상 코딩교

육을 하게 된다. 이에 따라 교육현장에서는 교육자 연수와 함께 다양한 교육도구를 이용한 교육 방법론에 대한 의견들이 제시되고 있다. 그러나 대부분 스크래치(엔트리)를 이용한 블록코딩과 아두이노 피지컬 컴퓨팅 교육이 주를 이루어 대학 교육에서 실무 업무에 필요로 하는 수학 알고리즘 기반의 논리적 사고력과 창의적인 문제해결 능력을 갖춘 인력양성이 어려운 형편이다.

본 고에서는 Raptor 순서도를 활용한 수학 알고리즘 기반의 코딩 교육방법을 제시하고, 645명을 대상으로 조사한 설문을 분석하였다. 먼저, 교육과정별로 바람직한 코딩 교육도구(프로그래밍 언어), Raptor의 유용성, 교육 실시 시기 및 주안점을 분석하고, 코딩 교육을 받았거나 개발 경력이 있는 사람과 그렇지 않은 비경력자들 사이의 유의성을 분석하였다. 분석결과, 경력자와 비경력자 사이의 의견의 차이는 존재하나, 초등학교에서는 스크래치(엔트리)를 이용한 블록코딩을 가장 선호(43.4%)하였고, 중학교에서는 이 비율이 15.7%(블록코딩)로 감소하였으며, C/C++(15%)를 2순위로 선택하였다. 고등학교와 대학에서는 C/C++, 자바, 파이선, 비주얼베이직을 순서적으로 선호하였으며, 이 결과는 TIOBE와 Redmonk사 등에서 발표하는 전 세계 프로그래밍 개발 언어의 점유율 순위와 일치하였다.

Raptor 플랫폼은 미국 공군사관학교에서 개발하여 코딩 및 프로그래밍 기초 교육용으로 사용되고 있으며, ISO 기반의 순서도를 이용한 알고리즘 설계에 유용하다. 이에 대해 응답자의 93%가 유용성에 동감하였고, 특히 1~5년 사이, 초보자들에게서 긍정적 평가가 많았다. 대부분 응답자들은 초등학교보다 중학교(45%)부터의 교육을 원하며, 교육 시 주안점은 논리적 사고력(28.2%), 문제해결 능력(24.5%), 창의성(24.2%)을 꼽았다. 특히, 주안점에 대한 이러한 결과는 코딩 언어문법(14.9%) 보다 비율이 높아 현재 흥미위주와 따라하기 식의 블록쌓기 교육에 대한 방향을 전환하여 수학 알고리즘 기반의 코딩 교육모델을 새롭게 수립해야 함을 시사적으로 나타내고 있다.

다소 늦은 감이 있고, 교육시간이 많지는 않지만, 우리나라에서도 코딩이 정규교과목으로 들어온 것은 고무적이다. 하지만, 지금처럼 흥미위주의 교육 방법은 단순히 공식을 외우고 대입하여 문제를 푸는 수학 교육과 전혀 다를 게 없다. 이런 방법으로 코딩교육을 계속 실시하면, 수학을 포기한 '수포자'에 이어 코딩을 포기한 '코포자'를 길러낼 뿐이고 이들은 모두 논리적 사고력과 창의적인 문제해결 능력이 없는 '수코포자'가 될 것이다. 바람직한 소프트웨어 인력 양성을 위해 지금이라도 교육 및 소프트웨어 관련자들의 의견을 반영한 새로운 코딩 교육모델의 정립이 필요한 이유이다.

[참고문헌]

[1] 노형진, 정한열, 한글 SPSS 기초에서 응용까지, 형설출판사, 2018.
[2] 오미자, 김미량, "컴퓨팅 사고력 향상을 위한 스크래치 프로그래밍 교육의 효과 분석", 교육정보미디어연구, 제

24권, 제2호, 2018. 6., pp.255-276.
[3] 유명현, 김재현, 구요한, 송지훈,"VR, AR, MR 기반 학습의 효과에 관한 메타분석", 교육정보미디어연구, 제24권, 제3호, 2018. 9., pp.459-488.
[4] 이애화, "국내 소프트웨어 교육 연구동향 분석", 교육정보미디어연구, 제24권, 제2호, 2018. 6., pp.277-302.
[5] 장희선, "4차 산업혁명 시대의 코딩 교육을 위한 수학 알고리즘 교육과정 연구개발", Proceedings of Mathematics to Industry, Korea, May, 2018., pp.104-123.
[6] 장희선, "4차 산업혁명의 시사적 교육을 위한 e-NIE 및 Edmodo 콘텐츠 활용", 한국콘텐츠학회 2018 춘계종합학술대회, 목포대학교, 2018. 5., pp.369-370.
[7] 장희선, "대학 정보기술 교육 분야에서 뉴스·신문 읽기 강좌의 효과 분석", 교육과학연구, 제20권, 제1호, 제주대학교 교육과학연구소, 2018. 5., pp.1-26.
[8] 장희선, "Raptor와 가상현실 콘텐츠를 활용한 수학 알고리즘 및 코딩 교육", 한국콘텐츠학회 2018 Contents & E-book, 전남대학교, 2018. 10., pp.20-21.
[9] 정영식, 유정수, 임진숙, 손유경, 소프트웨어 교육론, 씨마스, 2015.
[10] CODE, https://code.org.
[11] CoSuDa, https://sites.google.com/view/cosuda.
[12] Entry, https://playentry.org.
[13] Hee-Sen Jang, Dong Chul Kim, "Research and Development of Mathematical Algorithm Curriculum for Coding Education in Industry 4.0 Era", Proceedings of 2017 International Conference of Joint Societies for Mathematics Education: KSME, KSESM, Singapore NIE(KSME Policy Committee for Mathematics Education), Korea, Dec. 2017., pp.549-553.
[14] Hee-Seon Jang, Jang Hyun Baek,"Performance Analysis of Registration Schemes in Mobile Communication Network", The 6th International Symposium on Advanced & Applied Convergence (ISAAC 2018), Korea, November 2018., pp.68-72.
[15] Hee-Seon Jang, "Mobility Modeling and Analysis in Mobile Communication Networks", The Tenth International Conference on Ubiquitous and Future Networks(ICUFN 2018), Prague, Czech Republic, July 2018., pp.641-643.
[16] Martin Carlisle, http://raptor.martincarlisle.com(Raptor download).
[17] RedMonk, https://redmonk.com. 2018. 1.
[18] Scratch, https://scratch.mit.edu.
[19] Steve Hadfield, Troy Weingart, Wayne Brown, An Introduction to Programming and Algorithmic Reasoning using Raptor, United States Air Force Academy, CreateSpace an Amazon.com Company, 2018.
[20] Stewart Venit, Elizabeth Drake, Prelude to Programming Concepts and Design, PEARSON, 2015.
[21] TIOBE Index, https://www.tiobe.com, 2018. 11.
[22] TISTORY, 2018년 프로그래밍 언어 순위 및 비교, http://untitledblog.tistory.com, 2018. 5.
[23] X.Q. Cheng, Visualized Computation(可視化計算), Tsinghua University, 2013.