

# RAPTOR 프로그래밍 교육도구를 이용한 알고리즘 교육

김성율<sup>†</sup> · 이종연<sup>††</sup>

## 요 약

소프트웨어 교육은 건전한 정보윤리의식을 바탕으로 컴퓨팅 사고력 기반의 문제해결력을 기르는데 목적을 두고 있다. 이러한 목적을 달성하기 위해서 많은 교육기관에서 EPL(Educational Programming Language), 피지컬 컴퓨팅, 로봇 등 다양한 교육이 시도되고 있다. 하지만 특정 EPL과 상용 제품 등에 초점이 맞추어진 교육은 컴퓨팅 사고력 향상에 초점을 맞추고 있는 소프트웨어 교육의 창의성 교육을 확실히 유도할 우려가 있다. 따라서 본 논문에서는 순서도 기반의 비주얼 프로그래밍인 RAPTOR를 알고리즘 교육 도구로 제안하였다. 또한 일반계고등학교 학생을 대상으로 5단계에 걸쳐 12차시의 교육을 적용한 결과 RAPTOR를 이용한 알고리즘 교육이 높은 효과가 있음을 확인하였다.

**주제어** : RAPTOR 소프트웨어, 컴퓨팅 사고력, 컴퓨터 교육, 알고리즘 교육, 프로그래밍 교육

## Education of Algorithms Using the RAPTOR Programming Educational Tool

SungYul KIM<sup>†</sup> · JongYun LEE<sup>††</sup>

### ABSTRACT

The main aim in software education is to improve problem-solving ability based on computational thinking with the healthy information ethics. For this purpose, many institutions have attempted various educational programs such as Educational Programming Language, Physical Computing, and Robot education. However, it is possible to obscure the essence of computer education for computational thinking if the computer education focuses on using certain special education programming language and products. Therefore, this paper suggests a method of algorithm education using RAPTOR which is a visual programming development environment and is based on flowcharts. In order to verify the effectiveness of the algorithms education using the RAPTOR, 16 high-school students were applied to an educational program for twelve hours on five steps and then we obtained positive results.

**Keywords** : RAPTOR software, Computational thinking, Computer education, Algorithms Education, Programming Education.

---

<sup>†</sup> 정 회 원: 동아고등학교 교사 김성율  
<sup>††</sup> 종신회원: 충북대학교 교수 이종연(교신지자)  
 논문접수: 2015년 9월 16일, 심사완료: 2015년 11월 6일, 게재확정: 2015년 11월 13일

## 1. 서론

소프트웨어 산업의 부가가치 증대와 교육 선진국의 소프트웨어 교육 강화 등 시대적 요구에 따라서 2015 문·이과 통합형 교육과정에 기존의 정보 교육과정을 소프트웨어 교육 중심으로 개편하여 초·중·고등학교에서 소프트웨어 교육을 강화할 예정이다. 교육과정 개정 방향에 따르면 소프트웨어 교육은 건전한 정보윤리의식을 바탕으로 컴퓨팅 사고력 기반의 문제해결력을 기르는데 기본 목적으로 두고 있으며, 이를 토대로 다양한 분야에서 소프트웨어 역량을 융합하여 창의적으로 문제를 해결할 수 있는 창의·융합 인재 양성을 목적으로 하고 있다.[1]

그리고 교육부의 소프트웨어 운영 지침[1]에 따르면 이러한 소프트웨어 교육의 목적을 달성하기 위해 ‘생활과 소프트웨어’, ‘알고리즘과 프로그래밍’, ‘컴퓨팅과 문제해결’의 3가지 영역으로 초·중학교 중심의 교육과정을 제시하고 있다.

하지만 교육부에서 제시한 소프트웨어 운영 지침은 초·중학교를 대상으로 하고 있으며, 고등학교와 관련된 내용은 학교급간 내용 연계를 이해하기 위한 자료로 제시하고 있는 실정이다. 또한 2015 문·이과 통합형 교육과정에서는 소프트웨어 교육을 초등학교 실과 영역에 17시간, 중학교 34시간 이상의 교육과정만을 필수교육과정으로 제시하였고, 고등학교에서는 심화선택에서 일반선택 교과로 제시함으로써 초·중·고등학교의 연계성을 가진 교육과정 제시가 이루어지지 못하였다. 특히 대학입시위주의 교육과정을 운영하는 일반계고등학교에서 대학입시에 반영되지 않는 소프트웨어 교과의 선택을 기대하기 어렵다는 점은 초·중·고등학교의 지속적이고 연계성 있는 성공적인 소프트웨어 교육이 가능할까라는 의문을 주기에 충분하다.

뿐만 아니라 현재 효과적인 소프트웨어 교육을 위한 다양한 시도가 학교 현장에서 이루어지고 있는데, 교육 도구로서 활용되어야 하는 EPL(Education Programming Language), 센서보드(Sensor Board)와 상용 로봇 제품 활용 교육이 소프트웨어 교육의 중심에 있는 듯한 분위기는 컴퓨팅 사고력(Computational Thinking)을 핵심

키워드로 하고 있는 소프트웨어 교육의 본질을 흐릴 수 있는 가능성이 충분하다. 특히 로봇 교육의 경우 로봇 작동 원리의 교육보다는 특정 상용 제품의 활용법 및 사용법 교육이 주를 이루고 있으며, 최근 모처에서 개최된 로봇경진대회 3:3 배틀 축구 종목에서 대회에 참가한 학생들이 거의 모두 유사한 로봇으로 대회에 참가하였다는 점은 이 주장을 뒷받침해주고 있다.

이에 소프트웨어 교육이 성공적으로 교육 목표를 달성하고 학교현장에 정착하기 위해 해결해야 할 과제를 요약하면 다음과 같다. 첫째, 학생들이 초·중등학교에서 학습한 소프트웨어 교육을 바탕으로 대학과 진로까지 연계할 수 있는 고등학교 교육 체계가 필요하다. 고등학교 2015 교육과정에서 소프트웨어 교육을 심화선택에서 일반선택으로 전환하였지만 대학입시 문제, 전공 교원 부족 문제 등 학교 현장에서 짧은 시간 내에 선택 비중의 확대를 기대하기는 어렵다. 이러한 문제는 동아리 활동, 방과 후 활동 등의 차선택으로 해결되어야 할 것이며, 소프트웨어 산업에 흥미를 가지고 진로를 고민하는 학생들에게 지속적인 소프트웨어 교육을 제공할 수 있는 방법이다. 둘째, 다양한 소프트웨어 교육도구의 활용이 필요하다. 확정 되었거나 필수인 도구는 아니지만 많은 시범학교와 연구학교, 선도학교 등에서 EPL로서는 Scratch, Entry, Rurple 등을 활용하고, 로봇 교육과 피지컬 컴퓨팅 영역에서도 특정 상용 제품의 사용이 두드러지게 많다. 이러한 교육도구의 효과성을 의심하는 것이 아니라, 특정 도구에 집중된 교육은 창의적인 창의성 교육보다는 획일적인 창의성 교육이 이루어 질 수 있다는 점에서 심각하게 고민되어야 한다.

이에 초·중학교에서 학습한 소프트웨어 교육의 지식과 창의성을 확대하여 고등학교 정규교과, 동아리 활동, 방과 후 활동 등에서 적용이 가능하며, 블록 기반 언어(Scratch, Entry 등)이 아닌 비주얼 프로그래밍 기반의 효과적인 알고리즘 교육 도구인 RAPTOR의 사용을 제안하고자 한다. RAPTOR는 컴퓨팅 사고력의 향상을 위한 알고리즘 교육을 위해 활용이 가능하며, 순서도 기반의 비주얼 프로그래밍으로 사용법 숙지가 용이하고, RAPTOR 온라인 사이트를 통해 무료로 사용이

가능한 소프트웨어이다. 또한 RAPTOR는 순서도와 의사코드(Pseudocode)를 활용한 프로그램으로 초·중학교에서 학습한 EPL에서 C/C++/JAVA와 같은 범용 프로그래밍 언어로 진입하기 전 단계에서 효과적인 도구이며, RAPTOR를 통해 작성된 순서도는 실제 범용 프로그래밍 언어로 작성되었을 때와 똑같이 실행이 된다는 점에서 학생들의 흥미를 이끌어 내기에 충분하다.

따라서 본 논문에서는 RAPTOR의 교육적 가치를 제시하고 RAPTOR를 이용한 알고리즘 교육의 효과성을 검증하고자 한다.

## 2. 관련 연구

### 2.1 순서도를 활용한 알고리즘 교육

알고리즘(Algorithms)은 주어진 문제를 해결하는데 필요한 유한개의 명령어 집합(a set of instructions)라고 정의할 수 있으며, 알고리즘의 표현 방법에는 자연어, 순서도 의사코드(Pseudocode), 프로그래밍 언어 등이 있다. 2015 교육과정 소프트웨어 교육내용을 살펴보면 초·중학교에서 사용하는 알고리즘 표현 방법은 자연어로 기술하는 방법을 많이 사용하고 있다.

<표 1> Comparison of the existing visualization tools [8]

Feature	BACCII	FLINT	EC	FCI	Raptor	SFC	VL	IP	DF	A&Y	B#	PG	G&G	CVF
Year Published	1992	1999	2002	2003	2004	2004	2004	205	2006	2006	2006	2007	2007	2009
Flowchart	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Flowchart based programming	○	○	○	○	○	○	○	○	○	×	○	○	×	×
Flowchart generates code	○	×	×	×	○	○	○	○	○	×	○	×	×	×
Structural rules enforced	○	○	○	×	○	○	○	○	○	○	○	○	×	○
Colour used to fully differentiate components and structures	○	○	○	×	×	×	×	×	○	×	×	×	×	×
Code	○	×	×	×	○	○	○	○	○	○	○		○	○
Code based Programming	×	×	×	×	×	×	×	×	×	×	×	×	×	○
Code generates flowchart	×	×	×	×	×	×	×	×	×	×	×	×	×	○
Generated code compiles without modification	○	×	×	×	×	×	×	○	○	×	○	×	×	×
Code and Flowchart Displayed Concurrently	×	×	×	×	×	○	×	×	×	○	○	×	○	○
Synchronised highlighting of flowcharts and code	×	×	×	×	×	×	×	×	×	1/2	1/2	×	○	○
Synchronised Visual Execution of Flowcharts and code	×	×	×	×	×	×	×	×	×	○	×	×	○	×
Non visual execution	×	×	×	○	×	×	○	×	×	×	×	×	×	×
Visual execution	×	○	○	○	○	×	○	○	×	○	○	○	○	×
Variable inspector	○	○	○	○	○	×	×	○	×	○	○	×	×	×
Strongly typed	○	×	×	×	×	×	×	○	○	○	○	×	×	○
Error feedback	○	○	○	○	○	○	○	×	○	○	○	○	○	×
Java support	×	×	×	×	○	×	○	○	×	×	×	×	×	○
Variable	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Sequence	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Selection	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Iteration	○	○	○	○	○	○	○	○	○	○	○	○	×	○
Arrays	○	×	×	×	○	○	○	×	×	×	×	×	×	○
Procedures	○	○	×	×	○	×	○	×	×	×	×	×	×	○
Empirically Evaluated	○	○	×	×	○	×	○	×	×	×	○	×	×	×
Associated Pedagogy	×	×	×	×	×	×	×	×	×	×	×	○	○	×
OS Dependency	Win	Win	Win	Win	Win	Win	Win	Win	Win	Any	Win	Win	Win/Mac	Win

자연어를 활용하여 비구조화된 일상의 문제를 기술하는 학습도 주어진 문제를 해결하거나 절차를 세우는데 도움이 되겠지만, 순서도, 의사코드 등을 활용하여 비구조화된 문제를 구조화된 문제로 재구성 하는 교육도 효과적이다. 한옥영 외(2011)[2]는 정보교육을 위해서는 부분적 요소나 내용을 서로 관련지어 구조화시키는 것이 중요하며, 구조화는 알고리즘적 사고에 기초하고, 알고리즘적 사고에 기반한 문제 해결 능력 학습이 창의적인 문제 해결을 이룰 수 있다고 하였다.

순서도는 1921년 Frank B Gilbreth[4]에 의해서 ASME(The American Society of Mechanical Engineers)의 회원들에게 “PROCESS CHART”로 소개된 이후에 지금까지 알고리즘을 설명하는데 매우 효과적인 수단으로 사용되고 있는데, 특히 순서도 기반의 비주얼(Visual) 프로그래밍은 다양하게 활용되고 있다. IBM사의 웹 서비스 전략의 하나인 웹스피어(WebSphere)는 순서도 기반의 개발 환경을 제공하고 있으며[5], Tia Watts[6]가 개발한 C나 Pascal과 같은 구조의 텍스트 표현을 구현하고 순서도를 개발 할 수 있는 SFC가 대표적이다. 최근에는 매우 다양한 형태의 순서도 기반의 비주얼 프로그래밍 환경을 제공하는 도구가 등장하고 있다. <표 1>은 현재 사용되고 있는 순서도 기반의 비주얼 프로그래밍을 특징에 따라 비교한 것이다.

Scratch와 같은 교육용 프로그래밍 언어는 그 자체로 범용 프로그래밍 언어와 같이 컴퓨터가 이해할 수 있는 완벽한 언어라고 할 수 없다. 장운재 외(2011)[3]의 주장과 같이 정보(소프트웨어) 교과에서 단순한 언어기능에 대한 지식이 아니라 프로그래밍 언어를 통한 문제 해결력 향상에 알고리즘 교육의 중점을 두고 있다는 점에서 문법적이고 프로그램 자체의 어려움이 없는 순서도 기반의 비주얼 프로그래밍은 소프트웨어 교육에서 교육용 프로그래밍 언어보다 효과적이라고 할 수 있다.

## 2.2 RAPTOR

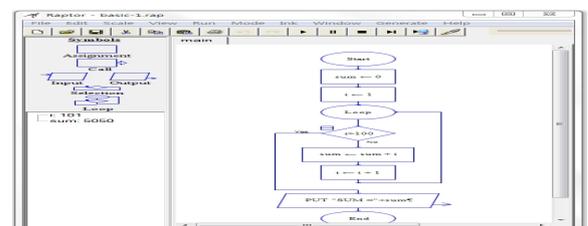
RAPTOR는 순서도에 기반 한 비주얼 프로그래밍(Visual Programming)으로 미국 공군 학교 교

육을 위해 개발되었으나, 현재 28개국 이상에서 컴퓨터 과학 교육을 위해 사용되고 있다. 하지만 국내 초·중등학교 교육 현장에서 컴퓨터 과학 교육 프로그램으로 우수한 가치가 있는 RAPTOR가 사용되는 사례는 거의 없는 실정이다.

RAPTOR는 비시각화된 환경과 어려운 구문이 가지는 단점의 극복과 정렬된 추론을 위하여 빠른 프로토타이핑(Rapid Prototyping)기법으로 설계되었다. 또한 학생들이 기본적인 그래픽 기호를 조합하여 알고리즘을 만들 수 있으며, 학생들이 만들어낸 알고리즘을 단계적으로 실행할 수 있다.[5] 뿐만 아니라 RAPTOR는 AdaGraph[7]에 기초한 단순한 그래픽 라이브러리를 제공하고 있어 학생들의 흥미를 유발하는데 매우 효과적이다.

Nikunja Swain P.E[8]는 <표 1>과 같은 다양한 순서도 기반의 비주얼 프로그램 중 RAPTOR가 논리적 사고력 교육 도구로 적합하다고 주장하였고, Michael Thompson[9]은 컴퓨터 과학 교육에서 RAPTOR를 활용한 교육의 효과성이 우수함을 실험을 통해 증명하기도 하였다. 그리고 Wayne Brown[10]은 RAPTOR가 프로그램 명령을 위한 최소한의 구문만 사용하며, 시각적이고, 사용이 쉬우며, 범용 프로그램의 이해 없이도 사용할 수 있다는 점에서 컴퓨터 과학 교육에 활용된다고 하였다.

RAPTOR는 <그림 1>과 같이 도구영역, 심볼영역, 작업영역, 확인창으로 구성이 되는데 사용자가 심볼영역에 있는 6가지의 기호를 작업영역으로 옮겨서 적절한 값(변수, 상수, 연산자 등)을 입력함으로써 임의의 문제를 해결할 수 있다.



<그림 1> RAPTOR 실행 화면

일반적인 순서도 작성 프로그램의 경우 순서도 기호를 활용해서 프로그램의 논리적 구조만을 보여주는데 그치는 반면, RAPTOR는 작성한 순서도를 사용자가 직접 실행시켜 봄으로써 프로그램

의 결과를 직관적으로 확인할 수 있다. 뿐만 아니라 복잡한 알고리즘을 단계별로 실행시키는 과정을 하이라이트(Highlight)를 통해 관찰할 수 있고, 사용자가 작성한 순서도를 Generate 기능을 통해 C#, C++, Java의 소스코드로 변환할 수 있다.

RAPTOR는 입력(Input), 처리(Processing), 출력(Output)의 3가지 기본 작업이 필요하며, 심볼(Symbol, 순서도 기호) 영역에서 원하는 심볼을 선택 후 작업 영역으로 드래그하여 간단히 사용할 수 있다. 그리고 각 심볼의 속성은 작업 영역으로 이동 후 더블클릭하거나, Edit 메뉴를 사용하여 손쉽게 작성할 수 있다. 또한 RAPTOR에서 사용되는 다양한 함수 설명은 RAPTOR 참고문서(Reference)를 통하여 즉시 확인할 수 있다.

### 3. RAPTOR를 활용한 수업 설계

#### 3.1 학습목표와 학습내용

본 논문은 RAPTOR를 활용한 효과적인 알고리즘 교육을 위한 4가지의 교육목표를 다음과 같이 선정하였다. 첫째, RAPTOR의 사용법을 이해하고 순서도를 작성할 수 있다. 둘째, RAPTOR를 이용하여 초보적인 수학문제를 표현하고 구현할 수 있다. 셋째, RAPTOR를 이용하여 자료구조를 순서도로 나타내고 구현할 수 있다. 넷째, RAPTOR를 이용하여 창의적 알고리즘 문제를 순서도로 나타내고 구현할 수 있다. 그리고 2가지 과정목표를 선정하였으며 다음과 같다. 첫째, 절차적 사고를 바탕으로 주어진 문제를 해결할 수 있다. 둘째, 소프트웨어(정보)의 중요성을 이해하고 적극적인 관련 진로를 탐색할 수 있다.

이러한 교육 목표를 달성하기 위하여 Start, Basic, Thinking, Creativity1, Creativity2의 5단계로 수업을 설계하였고, 각 단계별 학습내용은 다음과 같다. Start 단계의 학습 내용은 RAPTOR 기본 사용법 익히기와 간단한 순서도 그리기, Basic 단계의 학습 내용은 간단한 수학 문제의 해결, Thinking 단계의 학습 내용은 일반적인 수학 문제의 해결, Creativity1 단계의 학습 내용은 그래픽 프로그램의 표현, Creativity2 단계의 학습 내용은 그래픽 프로그램의 구현으로 선정하였다.

그리고 각 단계별 학생 활동 내용과 교사 활동 내용은 <그림 2>과 같이 설계하였다.

특히 학생 활동의 경우 RAPTOR를 이용하여 순서도를 직접 작성하기 전에 모듈별로 주어진 문제를 풀기위한 방법을 토의하고, 직접 찾게 함으로써 학생들의 적극적인 활동을 유도하였으며, 작성된 RAPTOR 프로그램을 학생들이 직접 다른 학생에게 설명하고 토의하는 방식의 평가를 진행하였다.

#### 3.2 교수학습지도안

RAPTOR를 활용한 효과적인 알고리즘 교육을 위해서 5단계(12차시) 교수-학습지도안을 작성하였으며, Start단계는 4차시(50분/1차시) 수업, 나머지 단계는 각각 2차시(50분/1차시) 수업을 진행하였다. 설계된 교수-학습지도안<sup>1)</sup>은 [부록 1]과 같으며 총 5단계로 작성되었다.

단계	학습내용	학생활동	교사활동
Start 단계	- Raptor기본 사용법 익히기 - 간단한 순서도 그리기	- 모듈별 5문항 문제해결방법 찾기 - 개인별 순서도 작성하기	지식전달 조언 평가
Basic 단계	- 간단한 수학문제의 해결	- 모듈별 4문항 문제해결방법 찾기 - 개인별 순서도 작성하기	조언 평가
Thinking 단계	- 일반적인 수학문제의 해결	- 모듈별 3문항 문제해결방법 찾기 - 개인별 순서도 작성하기	조언 평가
Creativity1 단계	- 그래픽 프로그램의 표현	- 모듈별 3문항 문제해결방법 찾기 - 개인별 순서도 작성하기	지식전달 조언 평가
Creativity2 단계	- 그래픽 프로그램의 구현	- 모듈별 2문항 문제해결방법 찾기 - 개인별 순서도 작성하기	조언 평가

<그림 2> 단계별 학습내용과 학생/교사 활동내용

### 4. 연구방법

#### 4.1 연구대상

알고리즘 교육 도구로서 RAPTOR의 효과성을 분석하기 위해 3절에서 언급한 것과 같이 총 5단계(Start - Basic - Thinking - Creativity1 - Creativity2)

의 교육 내용을 준비하였고, 교육대상은 D일반계 고등학교 1, 2학년 학생 중 컴퓨터 관련 학과로 대학 진학을 희망하는 자발적 참여 학생 16명을 선발하여 동아리를 구성하여 진행하였다.

### 4.2 실험방법

RAPTOR의 효과성을 분석하기 위해 연구대상자들에게 Start단계 4차시, Basic단계 2차시, Thinking단계 2차시, Creativity1 단계 2차시, Creativity2단계 2차시, 총 12차시 수업을 4주에 걸쳐 진행하였으며, 각 단계별 수업 내용은 <표 2>와 같다.

<표 2> 각 단계별 주요 학습내용

단계	차시	수업내용
Start	4차시	<ul style="list-style-type: none"> <li>- 순서도란 무엇인가?</li> <li>- 알고리즘이란 무엇인가?</li> <li>- 순서도 기호 익히기</li> <li>- Raptor란 무엇인가?(틱택톡 게임)</li> <li>- 4칙 연산 계산기 만들기</li> <li>- 두 좌표를 입력 받아 거리 구하기</li> <li>- 외부 파일을 통하여 시간표 만들기</li> <li>- 구구단 출력기 만들기</li> <li>- 서브차트를 이용하여 4칙 연산 계산기 만들기</li> <li>- 배열의 이해와 데이터 삽입</li> </ul>
Basic	2차시	<ul style="list-style-type: none"> <li>- 1부터 100까지 합을 구하는 프로그램 만들기</li> <li>- 임의의 두수를 입력 받아서 큰 수를 출력하는 프로그램 만들기</li> <li>- 3개의 숫자를 입력받고, 그 중 가장 큰 수를 출력하는 프로그램 만들기</li> <li>- 중간고사와 기말고사 성적을 입력 받아 학점 계산하는 프로그램 만들기</li> </ul>
Thinking	2차시	<ul style="list-style-type: none"> <li>- 임의의 두 정수를 입력 받아 두 정수 사이의 모든 정수들의 합을 구하는 프로그램 만들기</li> <li>- 0을 입력 받을 때까지 양의 정수를 입력 받아 합과 평균 계산하기</li> <li>- 수식을 만족하는 최대 정수와 합 구하기</li> </ul>
Creativity1	2차시	<ul style="list-style-type: none"> <li>- RaptorGraph 이해하기</li> <li>- 그래픽 윈도우 이해하기</li> <li>- 드로잉 명령어 이해하기</li> <li>- 사용자 인터페이스 이해하기</li> </ul>
Creativity2	2차시	<ul style="list-style-type: none"> <li>- 카운트다운 프로그램 만들기</li> <li>- 도형이 겹쳐지는 프로그램 만들기</li> </ul>

그리고 12차시에 걸쳐 RAPTOR를 활용한 알고리즘 교육을 실시하고, 그 효과성을 측정하기 위해 한국정보올림피아드 2015 시도예선 중 고등부 50개 문항 중 <그림 3>과 같이 C언어 프로그램을 활용하여 문제를 해결하는 문항 중 20개를 선

택하여 평가를 실시하였다.

RAPTOR가 단순히 도형만 활용하여 알고리즘을 표현하는 도구가 아니라 의사코드를 각 심볼(도형)에 입력해야 하기 때문에 C언어 프로그램을 활용한 문항이 적절하다고 판단하였다. 또한 한국정보올림피아드(KOI)는 미래창조과학부에서 주최하고 한국정보화진흥원에서 주관하는 국내 최고의 권위를 가진 대회이다. 또한 수학적 지식과 논리적 사고능력을 필요로 하는 알고리즘과 프로그램 작성 능력을 평가하기 때문에 한국정보올림피아드 2015 시도예선 중 고등부 문항은 학생들의 알고리즘 능력을 평가하는데 가장 효과적인 도구 중의 하나이다.

18. 다음 프로그램의 출력 결과는 무엇인가?

```

int n = 2015, i = 1, tot = 0;
int a = 0;
while (n > 0) {
    if (n % 10 == 2) {
        tot += (n / 10) * i + (a + 1);
    }
    else if (n % 10 > 2) {
        tot += (n / 10 + 1) * i;
    }
    else {
        tot += (n / 10) * i;
    }
    a += i * (n % 10);
    i *= 10;
    n /= 10;
}
printf("%d\n", tot);
        
```

① 616      ② 617      ③ 618      ④ 619      ⑤ 620

---

19. 다음 프로그램의 출력 결과는 무엇인가?

```

int n = 2015, c = 0;
while (n > 0) {
    c += (n & 3);
    n = n & (n - 1);
}
printf("%d\n", c);
        
```

① 0      ② 2      ③ 4      ④ 5      ⑤ 8

<그림 3> 평가 문항 일부

RAPTOR를 활용한 교육이 컴퓨팅 사고력의 하위 요소인 알고리즘 및 절차 영역에 효과가 있는지 측정하기 위하여 교육대상자 16명에게 사전, 사후 평가 후 그 결과를 활용하여 대응표본 t-검증을 실시하였는데, 사전 평가는 C언어 기초 문법을 활용한 예제 문제 교육 후 실시하였고, 사후 평가는 RAPTOR 프로그램을 적용 후 평가를 실시하였다.

대응 표본 t-검증은 모집단으로부터 추출된 동일 표본에 대해 프로그램의 효과가 있는지의 여부를 분석하는 방법으로 교육 프로그램의 효과성을 검증하는데 매우 효과적인 방법이다.

## 5. 연구결과

### 5.1 대응표본 통계량

<표 3>과 같이 사전검사 평균은 49.37, 사후검사 평균은 83.81로 약 34점의 차이가 발생하였다. 표준편차는 사전 검사는 13.64, 사후 검사는 9.51로 나타났다.

<표 3> 대응표본 통계량

	평균	N	표준편차	평균의 표준오차
사전검사	49.3750	16	13.6473	3.41184
사후검사	83.8125	16	9.51643	2.37911

### 5.2 대응표본 상관계수

<표 4>와 같이 사전검사와 사후검사 두 변수 간의 상관계수는 .910으로 상관계수가 높다고 할 수 있으며 유의확률 .000( $p < .05$ )이므로 통계적으로 유의하다. 또한 이 결과는 사전검사가 높은 교육대상자가 사후검사 점수도 높게 나타나는 경향이 있음을 나타내는 것이다.

<표 4> 대응표본 통계량

	평균	N	표준편차	평균의 표준오차
사전검사	49.3750	16	13.6473	3.41184
사후검사	83.8125	16	9.51643	2.37911

### 5.3 대응표본 t-검증

<표 5>과 같이 사전검사 점수와 사후검사 점수 간의 대응표본 검증 결과 t값이 -21.673이고 유의확률이 .000으로  $p < .001$ 이므로 통계적으로 커다란 유의미한 차이가 있다.

따라서 두 변수(사전검사, 사후검사) 간 평균 차이는 유의수준 .001에서 유의미한 차이가 있다고 할 수 있다.

그러므로 'RAPTOR를 이용한 알고리즘 교육은 효과가 있다'는 결론을 내릴 수 있다. 뿐만 아니라 기존에 교육 현장에서 많이 활용되고 있는 Scratch와 같은 교육용 프로그래밍 언어와 비교하여 RAPTOR는 순서도 작성을 통하여 하나의 완벽한 프로그램이 표현된다는 점에서 범용 프로그래밍 언어 교육을 위한 효과적인 교육 도구라고 할 수 있다.

<표 5> 대응표본 t-검증

사전 사후	대응차					t	자유 도	유의 확률 (양쪽 )
	평균	표준 편차	평균의 표준 오차	차이의 95%신뢰구간				
				하한	상한			
	-34.437 50	6.355 77	1.58894	-37.824 25	-31.050 75	-21.6 73	15	.000

## 6. 결론

컴퓨터(정보) 교과는 지금까지 크게 3차례의 변화를 겪어왔다. 첫 번째가 ICT 활용 중심의 교육 시대이다. 이 시기에는 컴퓨터가 보조 교과의 형태로 교과 정체성이 부족했고, 학교 현장에서 소외되는 등 많은 문제가 있었다. 두 번째는 컴퓨터 과학(Computer Science)의 교육이다. 하지만, 컴퓨터 과학의 경우 실생활과 괴리가 있으며, 난이도 조절에 실패를 하고 대학의 컴퓨터 전공 학과의 교육 내용을 축소한 형태로 성공을 거두었다고 주장하기 어렵다. 그리고 지금 세 번째 변화의 시작점에 서있다. 세 번째 변화는 소프트웨어 중심의 컴퓨팅 사고력이라는 핵심 키워드를 가지고 문제해결 능력을 신장시키기 위한 교육이다.

이 변화를 성공적으로 교육 현장에 정착시키기 위해서는 다양한 노력이 필요하며, 특히 과거에 우리가 실패를 경험했던 특별한 상용 프로그램 교육 중심의 교과 운영 등은 반드시 지양해야 한다. 하지만 앞으로의 소프트웨어 교육을 준비하는 지금 현 시점에서 여러 가지 교육 방법과 결과를 살펴보면 일부 프로그램과 특정 상용 제품이 활용이 두드러지게 높다는 것은 많은 고민을 해야 하는 부분이다.

본 논문에서 주장하는 것은 앞에서 지적한 문제점에 대한 대안이 RAPTOR를 활용한 교육이라는 것이 아니라, 특정 EPL과 특정 상용 제품의 사용도 교육적 의미가 있겠지만, 효과적이고 창의적인 소프트웨어 교육을 위해 의미 있는 교육적 도구들을 다양하게 활용하자는 것이다. 그러한 출발점에서 RAPTOR를 활용한 알고리즘 교육을 제안하였고, 앞으로도 컴퓨팅 사고력을 증진시키기 위해서 다양한 교육 도구들에 대한 효과성 검증이 이루어진다면 보다 창의적인 창의력 향상 교

육이 이루어질 것으로 기대한다.

끝으로 소프트웨어 교육이 학교 현장에서 올바르게 정착되어 학생들의 창의적 사고력을 향상시키고 급변하는 디지털 사회에 효과적인 교과로 자리 잡기 위해 보다 다양한 교육과정과 교육방법의 연구가 지속되어야 할 것이다.

### 참고문헌

[1] 교육부(2015). **소프트웨어 교육 운영 지침**.

[2] 한옥영, 김재현(2011). **효과적인 알고리즘 교육을 위한 교수-학습 모형 개발**. 한국컴퓨터교육학회 논문지 제14권 제2호.

[3] 장운재, 김자미, 이원규(2011). **EPL을 활용한 프로그래밍 교육에서 문제해결 수준이 프로젝트 완성도에 미치는 영향**. 한국컴퓨터교육학회 논문지 제14권 제6호.

[4] Frank B. Gilbreth & L. M. Gilbreth(1921). *PROCESS CHART: First Steps in Finding The One Way To do Work*. The American Society of Mechanical Engineers.

[5] Martin C. Carlisle, Terry A. Wilson, Jeffrey W. Humphries, Steven M. Hadfield(2005). *RAPTOR: A Visual Programming Environment for Teaching Algorithmic Problem Solving*. United States Air Force Academy, Department of Computer Science.

[6] Watts, T.(2003). *SFC - A Structured Flow Chart Editor Version 3*. Faculty Poster SIGCSE 2003. Available at: <http://www.cs.sonoma.edu/~tiawatts/SFC/>.

[7] vanDijk, J. *AdaGraph*. Online Internet. Available: <http://ada95.com/jvandyk/adagraph.html>

[8] Nikunja Swain P.E.(2013). *RAPTOR - A Vehicle Enhance Logical Thinking*. American Society for Engineering Education.

[9] Michael Thompson(2012). *Evaluating the Use of Flowchart-based RAPTOR Programming in CS0*. Division of Mathematics, Engineering, and Computer Science, Loras College.

[10] Wayne Brown. *Handouts - Introduction to Programming with RAPTOR*. Available: <http://raptor.martincarlisle.com/>

### 후 주

1) 지면관계산 1단계 교수·학습지도안의 일부 내용만을 제시함.

#### [부록 1] 설계한 교수-학습 지도안

차시	1차시/5차시(200분)
학습 주제	Raptor의 사용법
교육 목표	- Raptor의 사용법을 이해할 수 있다. - Raptor를 이용하여 초보적인 수준의 문제를 Flowchart로 만들 수 있다.
학습 자료	- 유튜브 동영상(RandallBower) 9편 Introduction, Variables, Files and Strings, Graphics, Selection, Loops, Subcharts, Animation, Arrays
CS/CT 항목	<input type="checkbox"/> 자료수집/ 분석/ 표현 <input checked="" type="checkbox"/> 문제 분해 <input type="checkbox"/> 추상화 <input checked="" type="checkbox"/> 알고리즘과 절차 <input type="checkbox"/> 자동화 <input type="checkbox"/> 시뮬레이션 <input type="checkbox"/> 병렬화 <input type="checkbox"/> 컴퓨터 동작원리 <input type="checkbox"/> 정보구조화 <input type="checkbox"/> 프로그래밍 <input type="checkbox"/> 정보윤리 <input checked="" type="checkbox"/> CT기반 문제해결 <input type="checkbox"/> 기타 _____
학습 단계	교수 학습 활동

#### - Flowchart(순서도)란 무엇인가?

어떤 문제를 해결하기 위한 방법이나 절차를 알고리즘이라고 한다. 이러한 알고리즘을 기술하는 방법에는 '자연어', '순서도', '의사코드', '프로그래밍 언어' 등이 있다.

순서도(Flowchart)는 알고리즘을 약속된 기호로 나타낸 그림을 의미한다.

#### - Flowchart 기호?

기호	명칭	사용 용도	기호	명칭	사용 용도
	시작	시작, 종료, 데이터 입력, 출력		입출력	수치입력, 시작, 종료 표시
	연산자	수행할 연산, 알고리즘		판단기호	판단기호의 결과
	연결선	데이터의 입력, 출력		사분	서로 다른 입력을 하나로 합침
	흐름선	이전 단계의 종료와 다음 단계의 연결을 위한 기호		하위단계	연속적인 단계를 나타냄
	반복	기억된 순서로 작업을 반복하는 기호		카드 호출	연속적인 단계를 나타냄
	데이터입출력 기호	데이터의 입력, 출력		다스레터	문자열, 데이터의 입출력

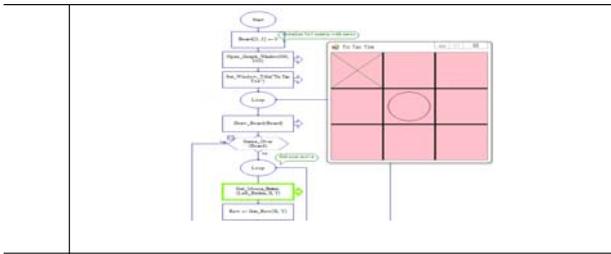
도입  
(30  
분)

#### - Raptor란 무엇인가?(프로그램 화면 제시)

Raptor는 Flowchart(순서도)에 기반 한 비주얼 프로그래밍(Visual Programming) 개발 환경이다. Raptor는 도구 상자, 심볼 영역, 확인 창(watch window), 작업 영역의 4개의 창으로 구성되고, 확장자는 ".rap"이다.

Raptor는 입력(Input), 처리(Processing), 출력(Output)의 3가지 기본 작업이 필요하다. 심볼 영역에서 원하는 심볼을 선택 후 작업 영역으로 드래그하여 사용한다. 작업 영역으로 이동된 심볼은 오른쪽 마우스를 클릭하여 Edit 메뉴를 이용하거나, 해당 심볼을 더블 클릭 함으로써 대화 상자를 선택할 수 있다.

#### - Raptor 사용 예시(Raptor samples - Raptor로 제작된 탁월 게임 일부)

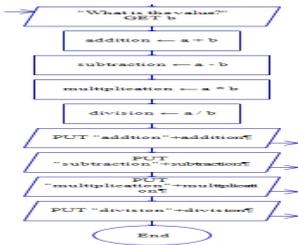


- 유튜브 동영상 시청
- + Introduction(6분33초)
- + Variables(5분19초)
- + Files and Strings(5분26초)
- + Graphics(3분18초)
- + Selection(5분)
- + Loops(7분18초)
- + Subcharts(6분9초)
- + Animation(5분12초)
- + Arrays(6분5초)

탐구  
(50  
분)

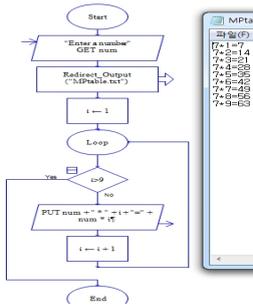


- 나의 첫 Raptor 프로그램(선생님 따라 하기 - 4칙 연산 계산기 만들기)



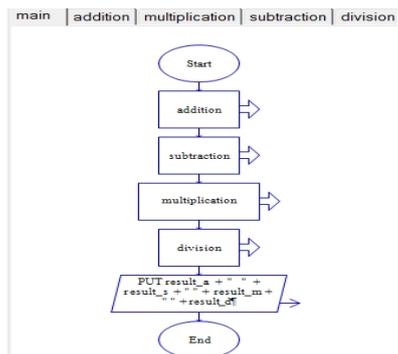
- Raptor 기능 익히기

예제) 구구단 중 원하는 단을 입력하면 파일로 출력하는 프로그램 만들기



학습  
활동  
(10  
0분)

예제) 서브차트(Sub Chart)를 이용한 사칙연산 계산기 만들기



## 김성을

2002 동아대학교 금속공학과 (공학사)  
2005 동아대학교 전자상거래학 (공학석사)

2010 충북대학교 정보·컴퓨터교육(교육학석사)  
2012 충북대학교 컴퓨터교육과(교육학박사)  
2002 ~ 2011 부산 동아중학교 교사  
2012 ~ 현재 부산 동아고등학교 교사  
관심분야: SW교육, 정보윤리, 정보교육  
E-Mail: ksungyul@hanmail.net



## 이종연

1985 충북대학교 전자계산기공학과(공학사)  
1987 충북대학교 전자계산기공학과 (공학석사)

1999 충북대학교 전자계산학과(이학박사)  
1990 ~ 1996 현대전자산업(주) 소프트웨어연구소  
현대정보기술(주) CIM사업부 책임연구원  
1999 ~ 2003 강원대학교 삼척캠퍼스 정보통신공학과 조교수  
2003 ~ 현재 충북대학교 소프트웨어학과 및 컴퓨터교육과 교수  
2001 ~ 2009 IEEE member  
2003 ~ 2004 한국정보처리학회 논문지편집위원 데이터베이스분과 이사 역임  
2010 ~ 현재 한국컴퓨터교육학회 이사(현)  
2010 ~ 현재 한국융합학회장(현)

관심분야: 데이터베이스, Biomedical Databases, 질의처리 및 최적화, 데이터 마이닝, e-Learning과 평가방법론, 정보과학영재 교육.

E-Mail: jongyun@chungbuk.ac.kr