

시각적 문해력을 활용한 프로그래밍 교육의 효과 : 예술계열 중심으로

Effects of Programming Education using Visual Literacy: Focus on Arts Major

피수영¹, 손현숙^{2*}

¹대구가톨릭대학교 컴퓨터소프트웨어학부, ²대구가톨릭대학교 SW중심대학

Su-Young Pi¹, Hyun-Sook Son^{2*}

¹Dept. of Computer Software, Daegu Catholic University, Daegu 38430, Korea

²National Center of Excellence in Software, Daegu Catholic University, Daegu 38430, Korea

[요약]

최근 소프트웨어 역량이 강조됨에 따라 대학에서는 전공 구분없이 모든 학생들에게 소프트웨어 교육을 진행하고 있다. 비전공자들은 소프트웨어 교육에 대한 동기부여가 부족하고, 생소한 학습 콘텐츠에 대한 체감 난이도가 높은 문제가 있다. 이를 해결하기 위해 학습자 특성에 맞는 소프트웨어 교육을 제공해야 한다. 예술계열 학생들은 시각에 대한 이해와 표현력이 뛰어나므로 시각적 문해력을 활용하면 프로그래밍 교육의 학습 효과를 높일 수 있다고 본다. 본 연구에서는 일상의 문제를 순서도와 의사코드로 분해하여 절차적으로 시각적 이미지를 구성한다. 이를 교육용 프로그래밍 언어인 플레이봇을 이용하여 코딩을 하고 문제를 해결하도록 진행하여 수업의 효과를 분석하고자 한다. 이를 통해 학생들은 프로그래밍의 개념을 이해하고, 문제를 컴퓨팅적 사고로 해결하는 과정을 이해할 수 있으며, 프로그래밍을 자신의 전공에 활용하는 방법을 습득할 수 있을 것으로 기대한다.

[Abstract]

Recently, with an emphasis on software proficiency, universities are providing software education to all students regardless of their majors. However, non-majors often lack motivation for software education and perceive the unfamiliar learning content as more challenging. To address this issue, tailored software education according to the learners' characteristics is essential. Art students, for instance, with their strong visual comprehension and expressive abilities, can benefit from utilizing visual literacy to enhance the effectiveness of programming education. In this study, we propose decomposing everyday problems into flowcharts and pseudocode to construct procedural and visual images. Using the educational programming language PlayBot, we aim to analyze the effectiveness of teaching by coding to solve problems. Through this approach, students are expected to grasp programming concepts, understand problem-solving processes through computational thinking, and acquire skills to apply programming in their respective fields.

Key Words: Flow Chart, PlayBot, Pseudo Code, Programming Education, Visual Literacy

<http://dx.doi.org/10.14702/JPEE.2024.105>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 2 March 2024; **Revised** 11 March 2024

Accepted 17 March 2024

***Corresponding Author**

E-mail: son10949@cu.ac.kr

I. 서론

현재의 디지털 시대에서는 소프트웨어가 산업과 사회의 거의 모든 영역에 연계되어 있어서 소프트웨어 교육의 중요성이 더욱더 강조되고 있다. 소프트웨어 교육은 코딩능력을 향상시키는 것 이상으로, 대학생들이 문제해결 능력, 협업 능력, 창의성 등을 함양하여 미래 사회에서 요구되는 역량을 갖추고 경쟁력을 확보할 수 있도록 도와준다. 소프트웨어 중심대학 중심으로 대부분의 대학에서 전공구분 없이 모든 학생들에게 소프트웨어 교육을 진행하고 있다. 대부분의 대학의 교양과정에서는 논리력, 창의적 문제해결 능력 그리고 컴퓨팅 사고력을 향상시키기 위해 소프트웨어 교육 중 프로그래밍 교육을 진행하고 있다[1,2]. 이러한 교육을 통해 비전공자들도 컴퓨팅 사고력을 향상시킬 수 있지만, 보다 효과적으로 이를 실현하기 위해서는 다양한 교육 접근 방식을 융합하는 것이 필요하다. 컴퓨터 비전공자를 대상으로 한 소프트웨어 교육은 비전공자에 맞는 교육목표와 교육방법 등을 고려하지 않고, 컴퓨터 전공 학생들을 위한 프로그래밍 중심의 교육방식을 그대로 적용하고 있다. 이러한 접근으로 인해 비전공자들이 소프트웨어 교육과정에서 학습하는데 많은 어려움을 호소하고 있다[3,4]. 비전공자들은 고등학교 교육내용과 전공 교육과정에서 상대적으로 수학적 사고 학습 비중이 낮고 절차적 문제해결 과정보다는 통합적 문제분석 훈련에 집중해 왔기 때문에 문제해결을 위한 사고체계와 전개과정이 컴퓨팅사고와는 차이가 있을 수 있다[5].

비전공자 대상으로 다양한 소프트웨어 교육이 활발히 이루어지고 있어, 비전공자 관련 소프트웨어 교육의 어려움에 관한 연구에 대한 관심도 증가하고 있다. 자연공학계열보다는 인문사회계열과 예체능계열 학생들이 더욱더 생소하고 어렵다고 응답을 했으며 흥미 향상, 사고력 향상, 성취도 향상 등의 항목에서도 자연공학 계열은 인문사회계열 및 예체능계열과 유의미한 차이를 보였다고 언급하였다[2,3]. 실제로 소프트웨어 교육을 이수해야 하는 인문계열과 예체능계열 학습자 입장에서는 소프트웨어 기초교육이 어렵다는 의견과 본인 전공과 관련이 없다고 생각하므로 소프트웨어 교육에 대한 동기부여가 부족할 수 있다고 언급하였다[6,7]. 동기 부족의 원인으로는 비자발적 선택으로 이루어지는 소프트웨어 기초교육 자체에 대한 거부감과 비전공자들에게는 생소한 학습 콘텐츠가 상대적으로 높은 체감 난이도로 지목되었다[8]. 비전공자 학습자들 중에서 특히, 예체능 계열의 학습자들은 소프트웨어 교육을 이수하는 경우가 많지 않다. 예체능 계열의 학습자가 프로그래밍 교육을 이수하는 데 어

려움을 겪지 않기 위해서는 예체능 계열의 학습자에게 적합한 소프트웨어 교육을 제공해야 한다.

본 연구는 예술계열 학생들에게 어려운 프로그래밍 교육의 학습효과를 높이기 위해서 학습자의 특성에 맞는 시각적 문해력(Visual Literacy)을 통해 접근하고자 한다. 시각적 문해력은 21세기 필수적 역량으로 손꼽힌다[9]. 시각화가 소프트웨어 개발에서 핵심요소라고 주장하는 이론[10-12]에 근거하여 시각적 표현을 기반으로 한 프로그래밍 교육을 진행하고자 한다. 예술계열 학생들은 시각에 대한 이해와 표현력이 뛰어나므로 시각적 문해력을 활용하면 프로그래밍 교육의 학습효과를 높일 수 있다고 본다. 이를 통해 프로그래밍 개념을 시각적으로 이해하고 활용하는 방법을 강조하여 학습자들의 이해도와 흥미를 증진시킬 수 있다고 본다. 시각적 문해력은 컴퓨터 프로그래밍 분야에서 중요한 역할을 하고 있지만, 국내에서는 미술교육에서 다루지고 있으며, 컴퓨터 프로그래밍이나 컴퓨터과학 교육과의 관련성을 보여준 연구가 거의 없으며[13,14], 시각적 문해력이 프로그램을 이해하고 작성하는 코딩 기반의 문제해결 능력에 영향을 미친다고 한다[13]. 교수자가 학습자에게 맞는 다양한 학습방식을 채택하여 교육하면, 학습자들의 참여와 집중력을 높일 수 있다. 따라서 교육내용을 설계할 때 학습자의 선호도와 특성을 고려하는 것은 중요하다고 본다. 학습자들이 자신에게 맞는 학습 방식을 경험하면 학습에 더욱 적극적으로 참여할 가능성이 높아지며, 이는 학습의 효과를 높일 수 있을 것이다.

본 연구에서는 먼저 일상의 문제를 순서도(Flow Chart)와 의사코드(Pseudo Code)로 분해하여 절차적으로 시각적 이미지를 구성한다. 이를 교육용 프로그래밍 언어인 플레이봇(PlayBot)을 이용하여 코딩을 하고 문제를 해결하도록 진행하여 수업의 효과를 분석하고자 한다. 이를 통해 학생들은 코딩의 개념을 이해하고, 시각적 프로그래밍을 통해 자신의 아이디어를 구현하는 방법을 배울 수 있다. 또한, 디지털 스토리텔링 기반의 학습활동을 통해 시각적 문해력, 공감각적 의사소통, 멀티미디어 활용능력, 자기주도적 학습 등의 교육적 효과를 얻을 수 있다고 본다.

II. 이론적 배경

A. 시각적 문해력

시각적 문해력(Visual Literacy)은 사전적 의미로 정보를 이미지의 형태로 해석하고 협의하며 의미를 만들어 내는 능력이다. 문학에서 문자를 읽고 쓸 수 있는 능력인 문해력

(Literacy)이 우리의 생활과 맞물려 있는 시각문화와 함께 확장되어 미술교육에서 시각적 문해력이라는 용어를 사용하고 있다[15]. 시각적 문해력은 그림이 지닌 의미를 인식하는 능력으로 많은 연구에서 정의하고 있으며[16,17], 정보를 전달하고 획득하기 위한 도구로도 정의하고 있다[16]. 소프트웨어공학 영역에서는 보이지 않는 성격(Invincible)을 지닌 소프트웨어를 이해하기 위한 목적으로 그래픽적인 시각적 기법을 사용하고 있다[10]. 시각적 문해력은 21세기 필수적 역량으로 자리매김하고 있으며[9], 다양한 미디어 시대에 시각적 제품과 메시지를 올바르게 이해하고 인식하고 해석하고, 다른 사람들과 함께 제작, 분석, 평가 및 의사 소통하는 능력으로 정의된다[9,18]. 시각적 문해력에 대한 정의는 1969년 J. L. Debes[19]의 연구를 시점으로 하여, 2011년 미국 도서관 협회의 한 부서인 대학과 연구 도서관 협회(Association of College and Research Libraries: ACRL)의 정의에 이르렀다[9,19]. 초창기 시각적 문해력은 시각적 메시지 이해, 읽기, 사용, 해석, 평가로 정의되었다[13].

ACRL은 시각적 문해력을 7개의 표준으로 나누어 제시하였다. 첫째, 시각적 이미지의 성격과 범위를 식별하여 필요한 이미지를 결정하는 능력. 둘째, 필요한 이미지를 찾아내는 능력. 셋째, 이미지와 영상매체를 해석하고 분석하는 능력으로 제시하였다. 넷째, 이미지와 영상매체를 윤리적으로 만들고 활용하는 능력. 다섯째, 이미지를 평가하는 능력. 여섯째, 이미지를 효과적으로 활용하는 능력이며 마지막 일곱째, 효과적인 의사소통과 의미전달을 위해 의도적으로 예술적 디자인을 하며 미적으로 품격화 시키는 능력으로 구분하였다[20]. 그리고 시각적 문해력의 구성요소를 의식 있는 인지, 비판적인 수용, 적절한 활용, 능동적인 조형활동, 창의력의 촉진으로 나누어 설명하였다. 즉, 시각적 문해력은 오늘날 지배적인 시각적 환경에서 개개의 시각적인 조형요소와 의미를 인지하고, 비평적으로 수용하며, 적절하고 의미 있게 활용하고, 능동적이고 창의적으로 제작하는 능력들과 숙련된 기술들을 의미한다고 하였다[22]. 시각적 문해력의 개념정의를 종합해보면 시각적 문해력이란 여러 시각적 매체에 담긴 시각적 정보를 올바르게 이해하고 해석하며 비판적으로 분석·평가하고, 자신의 의도를 시각적으로 표현하며 의사 소통할 수 있는 종합적인 능력이라고 할 수 있다[20].

B. 시각적 프로그래밍 언어 플레이봇(PlayBot)

시각적 프로그래밍 언어는 프로그래밍을 시각적인 방식으로 수행할 수 있는 언어를 의미하며 일반적인 텍스트 기반 언어와는 달리 그래픽 요소나 아이콘 등을 사용하여 프로그

램을 작성하고 있으며, 쉬운 인터페이스와 직관적인 명령어를 제공하여 프로그래밍 과정에서 학습자의 인지적 부담을 덜어준다. 특히 플레이봇(PlayBot) 같은 프로그래밍 언어는 로봇 시뮬레이션을 통해 학습자들이 가상의 상황에서 문제를 해결하며 프로그래밍 능력을 향상시킬 수 있는 교육용 언어이다. 자바 스크립트 언어를 이용하여 C언어와 동일한 문법을 사용하며 HTML5로 개발되어 다양한 브라우저에서 사용이 가능하지만 크롬에서 사용하기를 권장한다. 이 플랫폼은 로봇의 동작을 제어하는 함수(move(), turn_left())와 물건 및 비퍼와 관련된 함수(pick_object(), pick_beeper()), 그리고 센서 함수(front_is_clear(), front_is_door()) 등을 제공하며 학습자들은 주어진 문제 상황에서 이러한 함수들을 활용하여 문제를 해결하며 이를 통해 분석적인 시각, 논리적 사고력, 그리고 문제해결 능력을 함께 향상시킬 수 있다.

블록기반 언어와 텍스트기반 언어 사이에는 큰 차이가 있어서 학생들이 블록기반 언어를 학습했다 하더라도 텍스트기반 언어를 학습할 때 어려움을 겪는데, 플레이봇은 블록기반 언어와 텍스트기반 언어 사이의 연결을 제공하여, 학생들이 블록기반 언어를 학습한 후에도 텍스트기반 언어를 학습할 때 어려움을 해소하는 교육언어로서 용이하다[21]. 또한 로봇을 활용하여 프로그래밍을 배우는 동시에, 시각적 정보를 이해하고 활용하는 능력을 기르는 기회를 제공한다. 로봇이 가상의 환경에서 움직이며 미로를 탐험하거나 물건을 주워서 재배치 등의 동작을 수행하는데, 이러한 상황은 학생들에게 다양한 시각적 정보를 제공한다. 또한 로봇이 이동하거나 물체를 처리할 때의 그래픽적 표현은 학생들이 시각적으로 문제를 이해하고 해결하는 과정을 돕는다. 따라서 로봇의 동작과 환경을 시각적으로 이해하고 조작하는 과정을 통해 학생들은 문제 해결에 필요한 다양한 시각적 요소를 학습한다.

III. 연구방법

A. 연구 대상자

본 연구의 연구 대상자는 경북소재 4년제 대학에서 2022년 2학기 예술계열 디자인관련 학과 2학년 학습자 31명 대상으로 진행하였다. 비전공자 대상 코딩수업으로 일주일에 한 번 실습실에서 수업을 진행하는 계열별 교양필수 교과목으로 2학기에 개설하여 운영된다. 예술계열 학습자의 선호도와 특성을 반영하여, 비전공자에게 적합한 교육용 언어이면서 시각적 프로그래밍 언어인 플레이봇 활용한 코딩 교육을 통해

소프트웨어교육의 필요성을 인지하고 실생활의 문제를 컴퓨팅적 사고로 해결할 수 있는 능력 함양이 교육목표이다. 전체 연구대상자의 구성은 플레이봇 언어를 접하지 않은 학습자들로 남학생 5명(16%), 여학생 26명(84%)이다. 학습효과를 분석하기 위해 SPSS 25 프로그램을 활용하였다.

B. 시각적 문해력을 활용한 프로그래밍 교육 콘텐츠 설계

본 연구에서는 실 세계의 문제를 컴퓨팅적 사고로 해결하

기 위한 기본적인 컴퓨터의 구조와 소프트웨어 개발과정에 대해서는 학습하지 않고 프로그래밍 교육을 진행하였다. 문제해결과 컴퓨팅적 사고 교과목은 1학년때 교양필수로 수강해야 하는 교과목으로 대부분의 학생들이 1학년때 수업을 하고 2학년이 되면 계열필수와 학과지정 과목을 이수해야 한다. 따라서 본 연구는 2학년 대상으로 진행되는 교과목이라 모든 학생들이 이수한 것을 전제로 하여 콘텐츠를 설계하였다. 매 주차 실세계와 컴퓨팅 환경 사이의 차이를 이해하기 위한 내용들은 간단하게 언급한 후 진행하도록 설계하였다.

표 1. 시각적 문해력 활용한 문제해결 과정 예시

Table 1. Example of problem-solving process utilizing visual literacy

문제	순서도	의사코드
<p>로봇이 숫자카드(비퍼)를 줍는 동작을 통해서 숫자를 확인하고 선택 정렬 알고리즘을 이용하여 배열에 저장된 수들을 오름차순으로 정렬하고 숫자카드(비퍼)를 내려놓는다.</p>		<ol style="list-style-type: none"> 1) 카드의 숫자를 저장할 배열을 만든다. 2) 앞으로 한칸 이동한 후, 비퍼의 개수를 저장할 변수 (count)를 0으로 초기화 한다. 3) 비퍼를 줍는다. 4) Count 값을 1씩 증가한다. 5) 현재 칸에 비퍼가 없을 때까지 3)~4)를 반복 6) Count에 들어있는 값을 배열에 추가한다. 7) 2)~6)을 5번 반복한다. 8) 정렬 알고리즘을 이용해서 배열에 저장된 카드번호를 순서대로 정렬한다. 9) 원래위치 (1,1)로 되돌아온다. 10) 앞으로 한칸씩 이동하면서 배열에 저장된 카드번호 만큼 비퍼를 내려 놓는다. 11) 숫자가 작은 카드부터 내려놓으며 10)를 5번 반복한다.
<pre>list=new Array() for (i=1 ; i<=5 ; i++) { move() count=0 while (on_beeper()) { pick_beeper() count++ } list.push(count) } ... repeat("turn_left()",2) repeat("move()",5) repeat("turn_left()",2) for (i=0 ; i<=4 ; i++) { move() for (j=1 ; j<=list[i] ; j++) { put_beeper() } } move()</pre>	<p>코드</p>	<p>실행 초기</p> <p>실행 결과</p>

15주 수업 중에서 1주차에서 7주차 수업은 일반 수업방식으로 진행을 하고, 9주차부터 15주차 수업은 시각적 문해력을 활용한 프로그래밍 수업방식으로 진행하여 수업의 효과를 분석하였다. 9주차에서 15주차까지 수업은 표 1에 예를 나타낸 것처럼 순서도와 의사코드를 활용하여 문제를 해결할 수 있도록 콘텐츠를 미리 설계하여 진행하였다. 학습자들에게 문제를 제시한 후 문제를 해결하기 위해 순서도와 의사코드를 이용하여 해결방법을 설명해 준 후 학습자들이 플레이봇 코드로 문제를 해결할 수 있도록 진행하였다.

1) 시각적 문해력 활용한 수업 설계

9주차부터 실생활의 문제를 해결하기 위해 문제를 순서도와 의사코드를 이용하여 절차적으로 시각화 하여 표현하는 방법을 먼저 학습한다. 순서도와 의사코드의 개념에 대해 알아본 후 간단한 문제를 통해 시각화 하여 표현하는 방법을 습득한다. 이를 컴퓨터로 해결하기 위해 플레이봇 언어로 코딩하여 문제해결 과정을 분해하고 상세하게 절차적으로 전개하는 과정을 학습하게 된다. 학습자들에게 순서도의 자주 사용하는 몇 개의 기호만 설명해 준 후 학생들 자신이 이해할 수 있도록 정해진 규칙이 아닌 자유로운 그림 형식으로 작성하도록 진행하였고, 순서도를 보면서 플레이봇 컴퓨팅 환경에 맞는 의사코드를 작성한 후 코딩을 하도록 진행하였다. 150분 수업 중 60분은 교수자가 그림 1에 표시한 단계별 문제해결과정을 통해 문제를 해결하는 것을 전체적으로 설명해 준 후 학습자들과 함께 문제를 해결한다. 나머지 수업은 교수자가 주차 별 미리 설계한 문제를 학생들에게 제시한 후 학습자들이 본인이 이해할 수 있는 방식의 순서도와 의사

코드를 작성한 후 그림 1의 단계별 과정을 거쳐 문제를 스스로 해결할 수 있도록 지도하였다.

순서도 작성시 시각적 문해력 활용하여 정보를 명확하게 전달하도록 했다. 이를 위해 첫째, 불필요한 세부사항을 제거하여 핵심적인 문제에 집중하도록 지도하였다. 둘째, 시각적인 요소를 활용하여 본인이 이해할 수 있는 그림형식으로 표현하는 것도 허용하였다. 정해진 규칙에 맞게 작성하는 것이 부담스럽고 처음 접하는 것이라, 본인이 이해할 수 있는 그림형식으로 작성하도록 진행하였다. 셋째, 각 단계나 절차에 대한 중요한 사항들은 명확한 설명을 제공하여 작성하도록 진행하였다. 작성한 순서도를 의사코드로 작성시 변수명, 함수명, 제어문 등의 표현을 일관성 있게 작성하도록 진행하여, 코드를 보다 읽기 쉽고 이해하기 쉽게 작성할 수 있도록 했다. 의문사항이나 질의를 실습시간에 자유롭게 교수자 혹은 다른 학습자들과 소통하면서 해결하도록 진행하였다. 해결된 문제는 결과보고서를 학습자가 작성하여 강의지원 과제란에 제출하여, 수업 시간에 학습한 내용을 전체적으로 한 번 더 인지할 수 있도록 진행하였다. 처음에는 결과보고서 작성하는 것이 귀찮고 힘들었지만, 결과보고서 작성을 통해 학습 목표를 달성할 수 있었고, 수업에 적극적으로 참여함과 동시에 학업 성취도에 영향을 미쳤다고 응답하였다. 개인별로 문제코딩을 효과적이고 효율적으로 학습하기 위한 사용자의 경험을 디자인하는 디지털 스토리텔링 기반의 학습활동을 통해 창의력과 시각적 문해력, 공감능력, 멀티미디어 활용능력, 자기주도적 학습 능력 등의 교육적 효과를 얻을 수 있다고 본다.

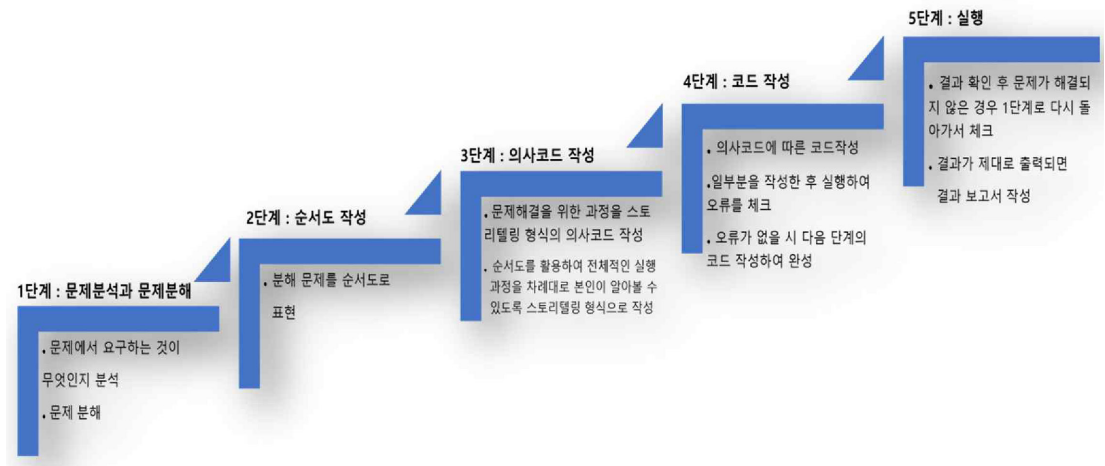


그림 1. 단계별 문제해결 과정

Fig. 1. Step by step problem solving process.

IV. 연구 결과

A. 설문조사와 중간평가 분석

1) 사전/사후 설문조사

학습자들의 소프트웨어 교육의 필요성 인식여부를 파악하기 위해 1주차 시에 간단한 설문조사를 네이버폼을 이용하여 실시하였다. 교양필수 교과목인 문제해결과 컴퓨팅적 사고를 수강한 적이 있는가?에 대한 응답은 29명(93.5%)이 수강한 것으로 나타났다. 2명(6.5%)이 수강하지 않은 것으로 나타났다. 소프트웨어 관련 기초 교과목인 문제해결과 컴퓨팅적 사고를 수강한 후 코딩교육을 수강하게 되면 컴퓨팅적 환경에 맞게 학습자들이 프로그래밍하는데 어려움이 없을 것으로 본다. 시각적 문해력에 대해 알고 있는가?에 대해서는 7명(22.5%)의 학생들이 알고 있다고 응답했다. 비전공자 학습자들도 프로그래밍 교육을 이수해야 한다고 생각하는가?에 대해서는 표 2에 나타난 것처럼 18명(58.1%)이 부정적으로 응답하였다. 수업을 진행한 후에는 6명(19.4%)이 부정정적으로 응답하였다. 12명(38.7%) 학습자들이 부정적에서 긍정적으로 인식변화가 일어났다. 수업을 수강하기 전에는 프로그래밍의 개념과 효과에 대한 인식이 부족하여 필요성을 인지하지 못하였지만, 실생활의 문제를 시각적 문해력 활용한 프로그래밍으로 해결하는 교육으로 인해 필요성을 인지하게 되었다고 응답하였다.

4차산업혁명시대에 자신의 전공에 IT를 연계하여 활용하는 교육이 필요하다고 생각하는가?에 대한 응답은 표 3에 나타난 것처럼 17명(54.8%)이 부정적으로 응답하였다. 수업 후

에는 4명(12.9%)이 부정적으로 응답하였다. 전공에 IT를 어떻게 연계시켜야 하는지 알 수 없어서 소프트웨어교육이 필요 없다고 생각했는데, 수업 후 어떻게 연계시켜 활용하는지를 알게 되어서 필요성을 인지하게 되었으며, 전공과 연계시켜 실생활의 문제를 해결할 수 있을 것 같다고 응답하였다.

2) 중간 평가

7주차부터 8주차에 걸쳐서 중간평가를 실시하였다. 대부분의 학생들이 프로그래밍 교육을 처음 접하여 생소하고 이해하기가 어렵다고 응답했다. 학습자들이 혼자서 문제해결을 위한 코딩 시 어려운 점을 크게 3가지로 분류하면 첫째, 변수사용 여부로 변수를 왜 사용해야 하는지 모르겠다고 응답했다. 이는 변수의 개념을 이해하지 못한 학습자들로 프로그래밍의 가장 기초가 되는 변수의 필요성에 대해 한번 더 언급해야 함을 평가를 통해 인지하게 되었다. 둘째, 문제를 해결할 알고리즘 사용 여부로 학습자들이 실생활의 문제를 컴퓨팅적 사고로 변환하여 문제를 해결하는데 어려움이 있는 경우로, 다양한 문제를 많이 해결함으로 인해 향상될 것으로 본다. 셋째, 어떤 제어문을 사용해야 하는지 여부로 특히 반복문의 범위와 종료에 대해 어려움이 있는 것으로 본다. 우리의 일상 생활에서 반복해야 할 문제들을 반복문으로 해결하는 다양한 예제들을 통해 좀더 쉽게 사용할 것으로 본다. 넷째, 오류발생 시 해결방법으로, 학습자들이 컴퓨팅적 사고로 문제를 해결하는 방법에 익숙하지 않아서 오류가 발생하면 다른 방법으로 문제를 해결할 생각없이 포기함으로 인해 문제해결 능력이 향상되지 않음과 동시에 프로그래밍을 어려워하는 것으로 본다. 오류문제를 메일과 강의지원시스템

표 2. 프로그래밍 교육이 비전공자에게도 필요하다고 생각하는가?

Table 2. Necessity of programming education for non-majors

		매우 그렇다	그렇다	보통	그렇지 않다	매우 그렇지 않다
사전	빈도	0명	5명	8명	8명	10명
	백분율	0%	16.1%	25.8%	25.8%	32.3%
사후	빈도	4명	10명	11명	4명	2명
	백분율	12.9%	32.3%	35.5%	12.9%	6.5%

표 3. 전공에 IT를 연계하여 활용하는 교육의 필요성?

Table 3. Necessity of education that utilizes IT in connection with majors

		매우 그렇다	그렇다	보통	그렇지 않다	매우 그렇지 않다
사전	빈도	1명	6명	7명	11명	6명
	백분율	3.2%	19.4%	25.8%	32.3%	19.4%
사후	빈도	5명	12명	10명	3명	1명
	백분율	16.1%	38.7%	32.3%	9.7%	3.2%

표 4. 프로그래밍 교육의 효과 : 대응표본 t-Test

Table 4. Effectiveness of programming education: Paired samples t-Test

	M	SD	t	p
사전	81.10	8.957	-2.643	.013
사후	83.23	10.723		

의 Q/A를 통해 교수자에게 질의하여 해결하든지 수업시간에 질의를 통하여 해결하도록 진행하였다. 중간평가를 통해 인지한 사항들을 참고하여 9주차에 간단한 예제를 통해 변수의 필요성과 제어문 사용방법에 대해 한번 더 언급을 한 후 시각적 문해력을 활용한 프로그래밍 교육을 진행하였다.

B. 시각적 문해력 활용한 프로그래밍 교육의 효과와 성취도 분석

시각적 문해력을 활용한 프로그래밍 교육의 효과를 분석하기 위해 사전 실시한 평가와 사후에 실시한 평가에 차이가 있는지 알아보기 위해 대응표본 t-Test를 실시하였다. 표 4에 나타난 것처럼 유의확률은 0.013으로 0.05보다 작으므로, 사전과 사후 변화의 차이가 있는 것으로 나타났다. 즉, 시각적 문해력을 활용한 프로그래밍 교육이 학습자들의 학습효과에 영향을 미치는 것으로 나타났다. 학습자들도 강의평가를 통해 여러 단계를 거쳐서 문제를 해결하는 과정이 처음에는 복잡하고 힘들었지만, 시각적 문해력을 활용한 프로그래밍 교육이 문제를 해결하는데 도움이 되었다고 응답했다.

마지막 주차에 설문조사를 실시하였는데, 일반적인 수업 진행과 시각적 문해력 활용한 수업 진행이 학습하는데 차이가 있었는가?의 질문에 26명(83.9%)이 차이가 있었다고 응답했다. 차이가 있다면 어떠한 수업진행 방법이 학습하는데 도움이 되었는가?에 대해 22명(84.6%)이 시각적 문해력 활

용한 수업이 도움이 되었다고 응답했다. 처음에는 모든 문제를 순서도와 의사코드를 이용하여 시각화로 변경한 후 코딩하는 방법이 번거롭고 어려웠지만 반복하여 작성하니 일반적인 진행방법보다 코딩하는데 도움이 되었다고 응답했다. 실생활의 복잡한 문제를 순서도와 의사코드로 변환하여 표현함으로써 문제를 분해하는 능력과 체계적 실행 능력이 향상된 것 같다고 응답했다. 코딩으로 문제를 해결할 때에도 일반적인 진행방법보다 시각적 문해력 활용한 코딩이 오류발생률이 적어 좀더 쉽게 해결할 수 있었다고 응답했으며, 실습시간에 개별적으로 문제를 작성한 후 결과보고서를 제출하는 것이 프로그래밍에 대한 이해도를 높이는데 도움이 되었다고 응답하였다. 학습자들에게는 시각적 문해력 활용한 프로그래밍 교육이 프로그래밍에 대한 이해도를 높이고, 프로그래밍을 자신의 전공에 활용하는 데 동기부여가 되었다고 본다.

기말고사 진행 전 강의평가와 간단한 설문조사를 통해 학습자들의 프로그래밍에 대한 인식의 변화를 알 수 있었다. 첫째, 비전공자 학습자들의 프로그래밍 교육의 필요성에 대한 인식이 변화되었다. 둘째, 프로그래밍 교육에 시각적 문해력 활용한 수업이 학습의 이해도를 높였으며, 디지털 스토리텔링 예제 작성을 통해 수업이 흥미로웠으며 비전공자도 실생활의 문제를 컴퓨터로 해결할 수 있음을 알게 되었다고 응답했다. 셋째, 전공에 IT연계하여 문제를 해결하는 방법에 대해서 동기부여가 되었다고 응답했다. 제품 디자인 혹은 서비스 디자인 분야에 IT 기술을 활용하여 사용자에게 적합한 서비스를 제공할 수 있다. 디자인관련 전공에 IT를 연계하는 것은 새로운 기회를 제공하는 것으로 새로운 비즈니스 모델을 개발하고 새로운 시장을 개척할 수 있을 것이다. 총 16개의 문항으로 구성된 교과목에 대한 성취도 평가를 개강과 종강시에 각각 진행하였다. 문항은 체계적 실행력과 정보활용 능력으로 분류하였다. 체계적 실행력의 하위 요소는 문제해결

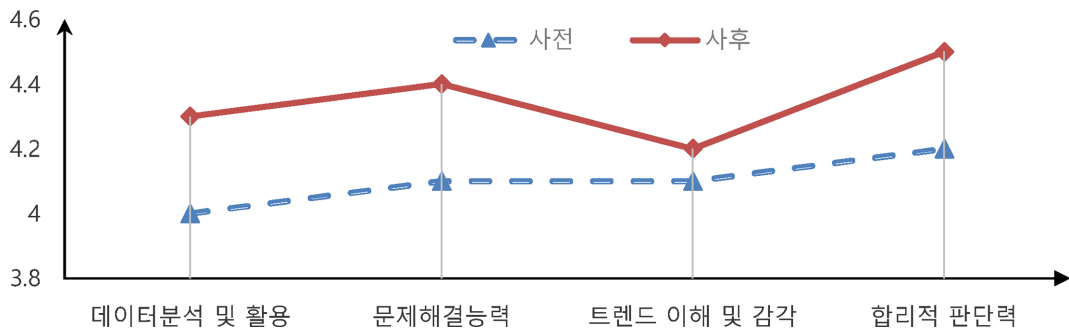


그림 2. 사전/사후 성취도 평가

Fig. 2. Pre/post achievement assessment.

능력과 합리적 판단력이며, 정보활용 능력은 데이터분석 및 활용과 트렌드 이해 및 감각으로 분류하여 설문하였다. 체계적 실행력은 사전 4.15에서 사후 4.45로 0.3 향상되었으며, 정보활용 능력은 사전 4.05에서 4.25로 0.2 향상되었다. 그림 2에 나타난 것처럼 하위 요소 영역 중 문제해결 능력과 합리적 판단력, 데이터분석 및 활용이 모두 0.3 향상되었음을 알 수 있었다. 실생활의 문제를 분석하여 순서도와 의사코드로 시각적으로 분석하여 절차적으로 표시한 후 이를 코딩으로 해결하는 과정을 통해 다양한 능력들이 향상된 것으로 본다.

V. 결론

최근 소프트웨어 역량이 강조됨에 따라 소프트웨어 교육의 필요성이 확대되고 있어, 대학에서는 전공 구분없이 모든 학생들에게 소프트웨어 교육을 진행하고 있다. 소프트웨어 교육을 통해 자신의 전공분야에 소프트웨어 기술을 적용하는 방법을 배우고 새로운 가치를 창출할 수 있으므로, 비전공자도 소프트웨어 교육을 통해 미래 사회에 필요한 역량을 갖추고 경쟁력을 확보할 수 있다. 비전공자에게 소프트웨어 교육은 단순히 기술적인 지식을 습득하는 것을 넘어, 미래 사회에 필요한 핵심역량을 키우고 전공분야와 융합하여 새로운 가치를 창출하는 데 중요한 역할을 한다. 따라서 다양한 교육 접근 방식을 융합하는 비전공자를 위한 프로그래밍 중심 교육과정을 개설하는 것이 필요하다. 컴퓨터 비전공자를 위한 소프트웨어 교육은 비전공자에 맞는 교육목표와 교육방법 등을 고려하지 않고 교육을 하고 있어 비전공자들이 소프트웨어 교육과정에서 학습의 많은 어려움을 호소하고 있다. 이를 해결하기 위해 학습자 특성에 맞는 소프트웨어 교육을 제공해야 한다. 예술계열 학습자들은 시각에 대한 이해와 표현력이 뛰어나므로 시각적 문해력을 활용하면 프로그래밍 교육의 학습효과를 높일 수 있다고 본다.

시각화가 소프트웨어 개발에 중요한 요소라고 주장하는 이론에 근거하여 시각적 문해력이 프로그래밍 교육에서 시각적 이미지를 구성하고 프로그래밍 하는데 어떠한 영향을 미치는지 수업의 효과를 분석하였다. 첫째, 프로그래밍 교육의 필요성에 대한 인식변화를 조사하기 위해 설문조사를 실시하였다. 그 결과, 수업 전에는 58.1%의 학생들이 프로그래밍 교육의 필요성을 부정적으로 인식했으나, 수업 후에는 19.4%로 감소하고 38.7%의 학생들이 긍정적인 인식으로 변화했다. 이러한 변화는 실생활 문제해결에 시각적 문해력 활용한 프로그래밍의 교육경험을 통해 프로그래밍의 실질적인 가치를 인지했기 때문으로 분석된다. 둘째, 4차 산업혁명 시

대에는 자신의 전공에 IT를 연계하여 활용하는 교육이 필수적이다. 사전 설문조사 결과, 과반수 이상(17명, 54.8%)의 학생들이 전공과 IT 연계 교육의 필요성을 인지하지 못했다. 이는 전공과 IT 연계 방법에 대한 지식 부족으로 인한 것으로 나타났다. 수업 후에는 긍정적인 변화가 일어났다. 13명(41.9%)의 학습자들은 전공과 IT 연계 방법을 알게 되면서 필요성을 인지하게 되었고, 전공과 연계하여 실생활의 문제를 해결할 수 있을 것으로 기대했다. 셋째, 시각적 문해력을 활용한 프로그래밍 교육의 효과를 분석하기 위해 대응표본 t-Test를 실시하였다. 그 결과 차이가 있는 것으로 나타났다. 즉, 시각적 문해력을 활용한 프로그래밍 교육이 학습자들의 학습효과에 영향을 미치는 것으로 나타났다. 학습자들도 강의평가를 통해 여러 단계를 거쳐서 문제를 해결하는 과정이 처음에는 복잡하고 힘들었지만, 시각적 문해력 활용한 프로그래밍 교육이 문제를 해결하는데 도움이 되었다고 응답했다. 넷째, 총 16개의 문항으로 구성된 교과목에 대한 성취도 평가를 진행한 결과, 체계적 실행력은 사전 4.15에서 사후 4.45로 0.3 향상되었으며, 정보활용 능력은 사전 4.05에서 4.25로 0.2 향상되었다. 실생활의 문제를 분석하여 순서도와 의사코드로 시각적으로 분석하여 절차적으로 표시한 후 이를 코딩으로 해결하는 과정을 통해 다양한 능력들이 향상된 것으로 본다. 학습자들에게는 시각적 문해력 활용한 프로그래밍 교육이 프로그래밍에 대한 이해도를 높이고, 프로그래밍을 자신의 전공에 활용하는 데 동기부여가 되었다고 본다. 향후 예술계열 학습자뿐만 아니라 다양한 계열 학습자들에게도 적용하여 학습자들의 학업성취도와 효과의 차이를 분석하여 다양한 융합교육 모델의 기반을 제공하고자 한다.

참고문헌

- [1] Y. S. Lee, "Python-based software education model for non-computer majors," *Journal of the Korea Convergence Society*, vol. 9, no. 3, pp. 73-78, 2018.
- [2] J. K. Kim and E. S. Sohn, "Difficulty analysis of an introductory computer programming course for non-major students," *Journal of Creative Information Culture*, vol. 7, no. 2, pp. 69-77, 2021.
- [3] K. S. Lee, "Case analysis for constructing a homogeneous learning group in programming lessons for non-specialists," *Journal of Digital Convergence*, vol. 17, no. 12, pp. 59-65, 2019.
- [4] J. S. Sung, S. H. Kim, and H. C. Kim, "Analysis of art and

- humanity major learners' features in programming class," *The Journal of Korean Association of Computer Education*, vol. 18, no. 3, pp. 25-35, 2015.
- [5] J. Y. Park, "Direction and prospect of SW education in the 2015 revised curriculum," *KEDI Journal of Educational Policy*, vol. 42, no. 3, pp. 85-89, 2015.
- [6] M. J. Lee, "Exploring the effect of SW programming curriculum and content development model for non-majors college students : focusing on visual representation of SW solutions," *Journal of Digital Contents Society*, vol. 18, no. 7, pp. 1313-1321, 2017.
- [7] S. H. Shin and J. Y. Seo, "Analysis of learner satisfaction by contents in basic software education of college of humanities," *Journal of the Korea Society of Computer and Information*, vol. 25, no. 6, pp. 251-261, 2020.
- [8] J. Y. Seo, "A case study on programming learning of non-SW majors for SW convergence education," *Journal of Digital Convergence*, vol. 15, no. 7, pp. 123-132, 2017.
- [9] D. Hattwig, K. Bussert, A. Medaille, and J. Burgess, "Visual literacy standards in higher education: New opportunities for libraries and student learning," *Libraries and the Academy*, vol. 13, no. 1, pp. 61-89, 2013.
- [10] R. Koschke, "Software Visualization in Software Maintenance," Reverse engineering, and re-engineering: a research survey," *Journal of Software: Evolution and Process*, vol. 15, no. 2, pp. 87-109, 2003.
- [11] E. Brumberger, "Visual literacy and the digital native: an examination of the millennial learner," *Journal of Visual Literacy*, vol. 30, no. 1, pp. 19-47, 2011.
- [12] D. L. Moody, "The physics of notations: A scientific approach to designing visual notations in software engineering," *ACM/IEEE 32nd International Conference on Software Engineering*, vol. 2, pp. 485-486, 2010.
- [13] C. J. Park and J. S. Hyun, "Analysis of the effects of learners' visual literacy and thinking patterns on program understanding and writing in basic coding education for computer non-majors," *The Journal of Korean Association of Computer Education*, vol. 23, no. 2, pp. 1-11, 2020.
- [14] C. J. Park and J. S. Hyun, "Visual literacy analysis affecting visual data design and coding of computer science subjects," *The Journal of Korean Association of Computer Education Summer Conference Proceedings*, vol. 23, no. 2, pp. 65-66, 2019.
- [15] H. H. Jang, "Development and application of teaching-learning program for improving computational thinking through visual communication education," M.A. Thesis, Seoul National University of Education, 2020.
- [16] P. A. Alberto, L. Fredrick, M. Hughes, L. McIntosh, and D. Cihak, "Components of visual literacy: Teaching logos," *Focus on Autism and Other Developmental Disabilities*, vol. 22, no. 4, pp. 234-243, 2007.
- [17] G. C. Roman and K. C. Cox, "Program visualization: The art of mapping programs to pictures," *Proceedings of the 14th International ACM Conference on Software Engineering*, pp. 412-420, 1992.
- [18] A. Kiper, S. Arslan, M. Kızılcı, and Ö. E. Akgün, "Visual literacy scale: The study of validity and reliability," *The Online Journal of New Horizons in Education*, vol. 2, no. 2, pp. 73-83, 2012.
- [19] K. Pem, "Enhancing high order science visual literacy skills in biology learners," Open University of Mauritius. Online Submission, 2019.
- [20] Y. O. Lee, "A study on the teaching method for developing visual literacy utilizing Synetics in picture books -Focusing on Pop-up book making activity," M.A. Thesis, Seoul National University of Education, 2016.
- [21] Playbot, "Education programming language," May 30, 2022. [Online]. Available: <http://playbot.spaceii.com/portal/about01.php>
- [22] Y. J. Hwang, "Teaching 'visual literacy' in art-education and understanding computer-animation," *Korean Elementary Art Education Association*, vol. 12, no. 1, pp. 183-197, 2001.



피수영 (Su-Young Pi) 정회원

2000년 8월 : 대구가톨릭대학교 전산통계학과(이학박사)
2012년 3월 ~ 현재 : 대구가톨릭대학교 컴퓨터소프트웨어학부 교수
<관심분야> 데이터마이닝, IT융합, 인공지능, 공학교육



손현숙 (Hyun-sook Son) 정회원

2001년 2월 : 대구가톨릭대학교 전산통계학과 박사
2019년 3월 ~ 현재 : 대구가톨릭대학교 SW중심대학 강의전담교수
<관심분야> 교육공학, SW교육, AI교육, 데이터 분석