



비전공자의 컴퓨팅 교양수업에서 MCPS 수업모형의 컴퓨팅 사고력 효과성 분석 사례 연구*

A Case Study on the Analysis of Effectiveness of Computational Thinking Ability in Non-major Computing Liberal Arts Class Based on the MCPS Instructional Model

박희정[†] · 전용주^{††}

Heejung Park[†] · Yongju Jeon^{††}

요약

본 연구는 비전공자를 대상으로 컴퓨팅 사고력 향상을 위해 학습동기 및 창의적 문제해결에 초점을 두고 컴퓨팅 교양수업을 설계하여 그 효과성을 분석하고자 하였다. 이를 위해 비전공자의 학습 동기 및 창의적 문제해결(MCPS) 수업모형 기반의 컴퓨팅 교양수업을 설계하고 이를 현장에 적용하였다. 컴퓨팅 사고력을 측정하기 위한 도구로는 정보과학에 대한 학습동기를 유발하고 컴퓨팅 사고력을 함양하기 위한 목적으로 개발된 비버챌린지를 활용하였으며 사전, 사후검사를 실시하였다. 또한 혼합방법연구 관점에서 MCPS 평가루브릭을 통해 수업 단계별 학생 성취도 평가 및 학생 강의평가 결과를 함께 확인하였다. 연구 결과 정량적 측면에서 비전공학생의 컴퓨팅 사고력 향상에서 유의한 변화를 관찰할 수 있었으며, 정성적 측면에서도 컴퓨팅 사고력 향상 뿐만 아니라 흥미, 성취감 및 자신감 향상 등 정의적인 관점에서 긍정적인 응답결과를 확인할 수 있었다. 이를 통해 MCPS 수업모형 기반의 컴퓨팅 수업은 비전공 학생들의 컴퓨팅 사고력을 향상시키는데 긍정적인 효과가 있음을 확인하였다. 본 연구의 결과가 향후 컴퓨팅 사고력을 함양하는 대학의 비전공자 컴퓨팅 교양 교육의 기초자료 및 교수법으로 활용되길 기대한다.

주제어 비전공자, 컴퓨팅 교육, 컴퓨팅 사고력, 학습동기, 창의적 문제해결, MCPS 수업모형

ABSTRACT

This study aimed to analyze the effectiveness of a computing liberal arts course designed to improve computational thinking ability for non-majors, with a focus on learning motivation and creative problem-solving skills. To achieve this, a computing liberal arts course based on the MCPS (Motivation and Creative Problem-Solving) instructional model was designed and implemented in the field. The Bebras Challenge tasks, developed to stimulate interest in information science and cultivate computational thinking ability, were used as a tool to measure computational thinking ability. Pre- and post-tests were conducted. Additionally, from a mixed methods research perspective, the results of student achievement evaluation and student lecture evaluation at each stage of class were confirmed through the MCPS evaluation rubric. The results showed significant improvements in the computational thinking ability of non-major students from a quantitative perspective. Qualitative data also indicated positive responses not only in terms of enhanced computational thinking ability but also in increased interest, sense of achievement, and confidence. These findings confirm that computing courses based on the MCPS instructional model have a positive effect on improving computational thinking ability among non-major students. It is hoped that the results of this study will serve as foundational data and teaching methods for future computing liberal arts education aimed at fostering computational thinking ability in university non-majors.

Keywords Non-majors, Computing education, Computational Thinking, Learning Motivation, Creative Problem Solving, MCPS Instructional Model

†정회원	국립안동대학교 교양교육원 강사
††중신회원	국립안동대학교 사범대학 컴퓨터교육과 부교수 (교신저자)
논문투고	2024년 06월 10일
심사완료	2024년 11월 05일
게재확정	2024년 11월 06일
발행일자	2024년 11월 20일

* 본 논문은 제1저자의 안동대학교 일반대학원 박사학위 논문 일부를 발췌하여 요약, 정리한 것임.

1. 서론

미래 사회를 능동적으로 준비할 수 있도록 전국민의 디지털 교육 기회 확대 및 역량을 강화하고 생활 문제와 융·복합적 문제를 해결할 수 있도록 각국에서는 정규 교육과정을 통해 컴퓨팅 사고력(Computational Thinking)을 갖춘 인재 양성을 추진 중이다. 이러한 시대적 흐름에 따라 국내에서도 국가 교육과정을 통해 초·중등 컴퓨팅 교육을 필수화하고 정보 시수를 확대하는 등 교육과정을 재구조화하고 있다[1, 2]. 대학에서는 컴퓨팅 사고력 강화를 위한 다양한 데이터과학 전공과 교과목이 신설되고 있다[3]. 대학의 교양 수준에서는 한국교양기초교육원에서 일반대학을 위한 '2022 교양 기초교육 표준모델'을 발표하였으며, 전문대학에 대해서는 '2024 전문대학 교양교육 권고안'을 통해 컴퓨팅 사고 교과목을 권고하고 있다[4, 5]. 또한 IT 기업이나 여러 비영리 단체에서도 컴퓨팅 사고력과 관련된 수업사례와 구체적인 교육 내용들을 다양하게 개발하고 있다[6].

그러나 대학의 교양 과정에서 이루어지고 있는 컴퓨팅 교육에는 비전공자의 특성을 고려하지 않고 기능 위주로 진행되거나 프로그래밍을 처음 접하는 학생들을 대상으로 기존의 강의 중심으로 진행되고 있어 중도 탈락, 흥미와 동기 부족 등 어려움을 호소하는 경우도 보고되고 있다[7-9]. 대학의 교양 과정에서 컴퓨팅 교육이 확산되고 있는 상황은 고무적이지만, 비전공자에게 적합한 프로그램은 여전히 부족하며, 비전공자의 컴퓨팅 사고력 함양을 위해 학생들에게 무엇을 어떻게 가르치고 평가할 것인지에 관한 효율적인 연구와 체계적인 분석 또한 필요한 상황이다[8].

최근 비전공자를 위한 컴퓨팅 교육의 효과성을 분석하는 연구들은 컴퓨팅 사고력 함양의 필요성과 그 방법론에 대한 다양한 접근을 제시하고 있다[3, 7, 10]. 컴퓨팅 사고력 기반의 프로그래밍 학습지원시스템을 개발하거나[10], 컴퓨팅 사고력 향상과 자신감 및 흥미를 고취시키는 수업을 설계하는 등 비전공자의 전공 및 단과대학 별 특성과 관심도에 따라 적절한 학습이 진행되어야 한다고 하였다[3, 7, 11]. 비전공자의 컴퓨팅 교육의 목적은 프로그래밍 언어의 문법을 학습하여 프로그램을 작성하는 방법을 배우는 것이 아니라 컴퓨팅의 기초적인 지식과 알고리즘에 대한 이해를 바탕으로 컴퓨팅 사고력과 논리력, 문제해결력을 함양하는 것이다[8, 9]. 대학의 교양교육에서 컴퓨팅 사고력 함양을 위해서는 컴퓨팅 사고력을 기반으로 일상생활의 문제를 창의적으로 해결할 수 있는 역량이 필요하므로[8], 컴퓨팅 사고력과 창의적 문제해결력 등 고등사고력에 중점을 두어 수업을 설계하는 것이 중요하다[6].

특히 비전공자의 컴퓨팅 사고력을 향상하기 위해서는 창의력과 문제해결력 향상에 초점을 두어야 한다[8]. 또한 동기 부여 없이 컴퓨팅 교육을 실시하게 되면 수업의 참여도와 교육의 효과가 약화되므로 동기 유발과 창의적 문제해결 역량을 함께 고려해야 할 필요가 있다[7, 12, 13]. 이에 더하여 컴퓨팅 교육에서 수업 전 학습자의 수준 진단 및 학습자의 어려움에 대한 분석이 선행되어야 하고[6], 컴퓨팅 사고

의 특징과 필요성에 대해 충분한 이해가 필요하며, 정서적 측면, 동기, 흥미 등을 위한 긍정적 피드백과 성찰 활동을 포함한 수업 설계와 교수법에 관한 연구가 필요하다[14].

따라서 비전공자의 컴퓨팅 사고력을 신장시키기 위한 컴퓨팅 교육은 사전에 학습자의 어려움을 진단하고 학습동기와 창의적 문제해결력을 함께 고려되어야 한다. 코딩 위주의 교육이 아니라, 프로그래밍 언어와 같은 도구를 이용하여 자료를 분석하고 적합한 알고리즘을 설계 및 구현하며, 컴퓨팅 기기를 이용하여 문제를 해결할 수 있도록 해야 한다[8-16].

한편 MCPS(Motivation and Creative Problem Solving) 수업모형은 대학의 비전공자를 대상으로 한 컴퓨팅 교육에서 학습자 특성을 고려하여 학습동기 및 창의적 문제해결력 향상을 위해 개발되었으므로 대학의 비전공자를 대상으로 한 컴퓨팅 교육에서 컴퓨팅 사고력을 향상하기 위한 교육적 접근에서도 가능성을 시사해 줄 수 있다[17].

따라서 본 연구에서는 비전공자의 컴퓨팅 사고력 향상을 위해 MCPS 수업모형 기반의 컴퓨팅 교양수업을 설계하고 이를 적용하여 컴퓨팅 사고력에 대한 효과성을 분석하고자 하였다.

2. 이론적 배경

2.1 컴퓨팅 사고력 함양을 위한 비전공자 교양교육 현황 및 관련 연구

컴퓨팅 사고력은 1980년 미국 MIT의 S.Papert 교수가 그의 저서에서 'computational thinking' 용어를 처음으로 사용하면서 도입된 이후, 미국 카네기멜론 대학의 J. Wing(2006, 2008) 교수가 그의 연구에서 추상화 및 자동화 등의 컴퓨팅을 활용한 사고능력 및 문제해결의 과정으로 재정의하면서 세계적인 주목을 받게 되었다. Wing(2006)은 컴퓨팅 사고력이 컴퓨터과학자의 사고방식이지만 비전공자들에게도 강조되어야 한다고 하면서 읽기, 쓰기, 셈하기(3R's)와 더불어 현대 사회를 살아가기 위한 필수 능력이라고도 주장하였다[18-20]. 이후 많은 연구자가 컴퓨팅 사고력을 재정의 하였으며, 이를 다양한 교육적 맥락에서 적용하기 위한 연구를 수행하고 있다[21, 22].

국외 대학의 컴퓨팅 교육을 살펴보면, 미국, 영국 그리고 유럽의 경우 대표적인 기관에서 대학의 컴퓨팅 교육을 주도하고 있다. 일본의 경우 정보처리학회에서 대학에서의 교양 교육으로서 요구되는 정보교육의 표준 커리큘럼을 제시하고 있다. 중동의 경우 대학의 컴퓨팅 교육은 ACM & IEEE의 커리큘럼을 따르고 있다. 이처럼 컴퓨팅 교육의 교육과정 및 교육 형태는 달라도 대학생에게 필요한 미래 역량을 길러주기 위해 각국에 맞게 활발히 진행되고 있다[17, 22].

국내대학의 경우 소프트웨어 중심대학을 필두로 컴퓨팅 교육이 확대되기 시작하였다. 각 소프트웨어 중심대학은 컴퓨팅 사고력 함양 및 컴퓨팅 교육의 효과를 높이기 위해

힘쓰고 있으며 비전공자 대상 컴퓨팅 교육이 교양 기초교육으로 자리매김하는데 선두 역할을 하게 되었다[17, 23]. 또 대학의 교양 기초교육 발전을 위한 한국 교양 기초교육원(2022)은 대학 교양 교육의 표준모델(2022)을 공표하기도 하였다[5].

컴퓨팅 사고력 기반의 컴퓨팅 교육 관련 연구를 살펴보면, 송옥지 외(2020)는 컴퓨팅 교육의 목표 중 하나는 컴퓨팅 사고력을 기르는 것에 있다고 하였다. 컴퓨팅 사고력을 기르기 위한 소프트웨어 교육을 위해 학교급별 과정에 맞는 교육방법이 중요하다고 하였다[24]. 김완섭(2019)은 대학생들의 컴퓨팅 교양필수 교과목에 관한 연구로 과거 프로그래밍 문법과 기법 중심에서 컴퓨팅 사고력 중심으로 그 방향성이 변경되었고, 대학의 교양필수 과목으로 컴퓨팅 사고력 관련 교과가 적합하다고 분석하였다[25]. 정혜진(2021)은 다양한 연구를 통해 컴퓨팅 기초교육 개발에 유의미한 성과를 거두고 있으나 컴퓨팅 기초교육 설계자가 의도한 교육적 효과성을 측정하는 연구는 부족한 편이라고 하였다[26].

한편 수업모형 개발 연구로 김영직(2020)은 컴퓨팅 사고력 기반 테스트 중심 문제해결(CT-TDPS) 수업모형 연구에서 모듈 프로그래밍이 컴퓨팅 사고력 스킬과 상관관계가 있으며 컴퓨터과학에 대한 충분한 학습동기와 프로그래밍 개념과 원리 및 모형에 대한 이해가 필요하다고 하였다[27]. 김효숙 외(2020)는 교수자의 시범을 따라만 하는 수업 방식은 프로그래밍 구현 및 기능 습득에 머무를 수 있다고 하면서, 컴퓨팅 사고력 역량과 프로그래밍 학습을 연결해 컴퓨팅 사고력 기반의 문제해결 중심 교수·학습 모형을 제시하였다[28].

이처럼 국내외에서는 컴퓨팅 사고력의 중요성을 인식하고 이를 신장시키기 위해 다양한 연구와 활동이 진행되고 있지만 여전히 대학의 비전공자 대상의 컴퓨팅 사고력을 신장시키기 위한 교육 프로그램과 교수법 및 효과성에 관한 연구는 부족한 실정이다. 또 효과적인 컴퓨팅 교육을 위해서는 컴퓨팅 핵심요소 뿐만 아니라 컴퓨팅 사고력 기반의 창의적 문제해결력, 학습동기, 컴퓨팅 기반의 소통 및 학습태도 등 다양한 변인들을 함께 고려할 것이다.

2.2 MCPS((Motivation and Creative Problem Solving) 수업모형

MCPS 수업모형은 비전공자를 위한 컴퓨팅 교양수업에서 학습동기와 창의적 문제해결력 향상을 위해 구체적인 교수학습 방법을 제시하는 수업모형이다[17]. 비전공자를 위해 개발된 MCPS 수업모형의 특징을 세 가지로 요약하면 다음과 같다. 첫째, MCPS 수업모형은 Papert의 구성주의 학습 원리를 바탕으로 개발되었다. 즉, 아이디어를 실체화할 수 있어야 하고, 결과물은 반드시 구체물을 바탕으로 구성하는 활동이어야 하며, 쉽고 재미있는 흥미로운 학습으로 이루어지도록 한다. 둘째, MCPS 수업모형의 개발 과정에서 비전공자 대상 컴퓨팅 수업의 문제점을 분석하

여 개선 방안을 적용하였다. 비전공자가 컴퓨팅 수업에서 겪는 문제점으로 학습동기 부족, 컴퓨터의 이해에 관한 선행 학습 필요, 프로그래밍의 어려움 등을 고찰하고 수업모형 개발에 반영하였다. 셋째, MCPS 수업모형은 비전공자의 학습효과를 높이기 위해 단계에 맞는 다양한 교수법의 원리를 반영하여 개발되었다. 또한 MCPS 수업모형에서는 학습자의 인지적, 정의적 측면에서의 학습 원리를 적용하였으며, 교수자의 수업 적용 과정에서 명확히 참고할 수 있도록 단계별 세부활동 및 전략을 함께 제시하고 있다[17].

MCPS 수업모형은 ‘동기부여 및 준비(Motivation)’ - ‘기초지식 형성(Building)’ - ‘기초지식 응용(Application)’ - ‘산출물 기획 및 제작(Production)’ - ‘발전(Development)’의 5단계로 구성되었다. MCPS 수업모형의 단계 및 내용은 Table 1과 같다.

Table 1. MCPS Instructional Model[17]

Steps	Details
Step 1. Motivation and Preparation	1.1 Learning motivation (relevance to major, need for computing education)
	1.2 Identifying learner's characteristics and difficulties in computing education (online survey)
	1.3 Identifying the antecedents of computing education
Step 2. Building of Basic Knowledge	2.1 Learning motivation (find related current IT issues)
	2.2 Demonstration (explain programming grammar)
	2.3 Support (using real-life examples)
	2.4 Stop helping (extend practice)
Step 3. Application of Basic Knowledge	3.1 Learning motivation (build expectations by introducing computing skills)
	3.2 Problem settings
	3.3 Derive a solution to the problem
	3.4 Share solutions
Step 4. Planning Outputs and Execution	4.1 Exploring ideas
	4.2 Ideation
	4.3 Design and share ideas
	4.4 Implementing ideas 4.4.1 Proofread ideas 4.4.2 Share concrete Outputs
	4.5 Evaluation
Step 5. Development	5.1 Learning motivation (identify challenges and confidence boosts)
	5.2 Applying digital technology
	5.3 Express ideas through various media

박희정(2023)은 비전공자의 컴퓨팅 교양수업을 위한 MCPS 수업모형 기반의 컴퓨팅 교육에서 학습동기의 내적 목표는 창의적 문제해결력의 특정 영역 지식·사고·기술 및 숙달과 확산적 사고에 영향을 미치는 데 있어 흥미·이해 기반 문제해결이 부분 매개역할을 하였으며, 디지털 기반 의사소통은 완전 매개역할을 하는 것을 확인한 바 있다.

또한 학습동기의 자기효능감은 창의적 문제해결력의 비판적·논리적 사고와 동기적 요소에 영향을 미치는 것을 확인하였다[17].

유지원(2022)은 학습자가 지닌 특징, 전공 배경, 숙련도를 고려하면서 소프트웨어 학습에 대한 가치를 인식하는 것이 동기적 요소와 함께 컴퓨팅 사고력 기반 문제해결력 과도 밀접하게 연관된다고 하였다. 또 앱인벤터 같은 블록 프로그래밍 교육 및 프로젝트 교육으로 컴퓨팅 사고력과 흥미가 증진되었다고 하였다[29].

고은지(2022)는 비전공자의 컴퓨팅 사고력을 위해서는 창의적이고 논리적으로 문제를 해결했는지에 초점을 두어야 할 필요가 있으므로 프로그램의 정확성 보다 학습자가 작성한 코드가 잘 실행되는지 판단하고 어떤 스타일로 코딩을 수행했는지 파악하는 것에 중점을 두어야 한다고 하였다[30].

이와 같이 MCPS 수업모형은 비전공 학습자의 가장 큰 문제점으로 지적되었던 학습동기와 컴퓨팅 사고력 함양에 필수적으로 고려되고 있는 창의적 문제해결력에 초점을 두고 개발되었다. 또한 인지적 측면 뿐 아니라 정의적 특성을 고려하여 컴퓨팅 수업 단계와 다양한 교수법을 포함하고 있어, 비전공자의 컴퓨팅 사고력을 함양하기 위한 컴퓨팅 교양수업에 적절한 모형이라 할 수 있다.

2.3 기존 모형과 MCPS 수업모형의 차별점

컴퓨팅 수업을 위한 기존 수업모형의 대부분은 초·중등 학생 대상의 연구가 주를 이루었으며, 한국교육개발원과 한국교육학술정보원에서 개발한 소프트웨어 교수·학습모형[31]을 기반으로 목적에 맞게 수정하여 사용되었다.

이에 반해 비전공자 대상의 창의적 문제해결력 및 학습동기를 위한 MCPS 수업모형은 기존 연구와 비교했을 때 다음과 같은 특징이 있다[17]. 첫째, Papert 구성주의 학습이론을 바탕으로 학습자 수준 및 학습자 속도를 조절하고 학습자 특성을 바탕으로 준비 단계를 구성하여 학습자에게 많은 선택 기회를 제공하고 반영함으로써 비전공 학습자가 만족감과 성취감을 느낄 수 있다.

둘째, 기존 연구모형의 대부분이 첫 번째 단계에만 동기 유발이 있으나 MCPS 수업모형은 단계 1에서는 전체적인 동기유발 활동이고 단계 2 ~ 단계 5까지 관련성 찾기와 관련 정보제공 등의 활동으로 각 단계별로 목표를 상기하고 진행 과정을 확인하는 등 모든 단계에서 학습동기 유지 및 강화할 수 있도록 하여 비전공자의 가장 큰 동기 부족 문제를 개선하였다.

셋째, 기존 수업모형은 대부분 단기간의 수업을 대상으로 적용되는 반면, MCPS 수업모형은 비전공자의 역량을 위한 효과적인 학습 원리를 적용하여 1학기 이상의 수업을 대상으로 적용할 수 있도록 구성하였다.

넷째, 기존 컴퓨팅 수업은 프로그래밍 학습 후 산출물 프로젝트를 과제로 수행하거나 산출물 결과 위주의 평가였으나 MCPS 수업모형은 기초지식 형성에서 응용 및 프로젝

트 산출물 기획 및 제작 과정 전체를 수업모형에 포함하여 체계적인 학습이 가능하도록 구성하였다.

다섯째, 기존의 산출물 제작 및 평가에 끝나지 않고 비교과 활동을 연계하여 도전 의식과 자신감을 고취하였다. 또 학내 행사 및 일정에 따라 선택할 수 있도록 구성하여 유동적으로 운영할 수 있다.

따라서 본 연구에서는 이러한 차별점을 중점으로 비전공자에게 어려운 컴퓨팅 사고력의 요소를 일일이 설명하여 나타내지 않아도 컴퓨팅 사고력의 요소를 수업의 단계에 적용하였다. 수업과 프로젝트를 진행함과 동시에 컴퓨팅 사고력을 기르는 과정을 포함해야 할 것이다.

3. MCPS 수업모형 기반의 수업 설계

3.1 프로그램 설계 방향

본 연구는 컴퓨팅 사고력 함양을 위해 학습동기와 창의적 문제해결력에 초점을 두고 MCPS 수업모형을 바탕으로 수업을 설계 및 산출물을 제작하는 과정으로 다음과 같은 방향성을 제시하였다.

단계 1은 동기부여 및 준비 단계로 본격적인 프로그래밍 수업에 앞서 학습자 특성 및 컴퓨팅 교육의 어려움을 파악한다. 컴퓨팅 교육의 중요성을 인식하여 전공과의 연관성 및 학습동기 부여가 이루어지도록 설계한다.

단계 2는 기초지식 형성 단계로 강의법 및 인지적 도제법의 학습 원리를 적용한다. 문법 설명 후 교수자가 시범을 보이고 학습자가 모방을 하면서 점차 도움을 줄여가며 사고를 확장해가는 방법으로 설계한다.

단계 3은 기초지식 응용 단계로 문제 기반 학습 및 개인 프로젝트 기반 학습 원리를 반영한다. 실생활 또는 계열(전공)관련 문제제시를 통해 스스로 해결안 도출 및 자기 성찰 활동을 통해 성공적인 경험이 가능하도록 학습 환경을 설계한다.

단계 4는 산출물 기획 및 제작 단계로 컴퓨팅 사고력 기반의 창의적 문제해결력(Computational Thinking-based Creative Problem Solving, CT-CPS) 및 팀 프로젝트 기반 학습의 원리를 반영한다. 산출물 제작을 위해 팀을 구성하고 아이디어 탐색, 아이디어 구상, 설계 및 공유, 구현 및 평가 과정을 통해 창의적으로 산출물을 완성하도록 설계한다. 이때 단계 2에서 학습하지 못한 필요한 문법은 예시를 통해 활용한다.

단계 5는 발전 단계로 시연 및 비교과 활동과 연계하도록 한다. 이를 통해 다양한 매체를 통해 아이디어를 표현하고 도전 의식과 자신감을 고취하도록 설계한다. 이때 단계 5는 생략할 수 있으며 생략 시 단계 3을 진행하도록 권장한다.

3.2 수업 설계

위 설계의 방향을 바탕으로 MCPS 수업모형 기반의 수업은 다음과 같이 설계하였다.

1 ~ 2주차는 전체과정에 대한 학습동기 부여 활동을 위해 컴퓨팅 교육의 필요성 및 전공 관련성 찾기 활동을 한다. 특히 1주차에서 학습자 특성 분석을 위한 온라인 설문을 진행하여 이후 수업에서는 학습자 수준과 학습 속도 및 과제 난이도를 조절한다. 이때, 실습 예제 및 과제의 난이도는 상, 중, 하로 준비하여 수업 상황에 맞게 진행한다. 또 실습 주제는 전공관련 내용으로 구성하여 동기유발과 친밀감을 느낄 수 있도록 한다. 과제는 생각을 확장하는 기능을 1~3개로 추가하여 과제 난이도를 조절한다.

3 ~ 8주차는 프로그래밍 문법을 학습하는 과정으로 학습자 특성을 파악하여 학습 수준과 학습 속도 및 과제 난이도 등을 고려하여 2가지 주제(예 : 변수와 조건문 이해, 조건문 이해와 반복문)를 중첩하여 제시한다. 또 매 수업 시작에는 최신 IT 이슈(쿠팡 물류센터 자동 분류 로봇 영상 시청, AI(Artificial Intelligence)가 그린 작품 우수상 논란 뉴스, 드라마 출연 배우 딥러닝 기술로 복원, AI가 디자인한 신발 및 의류 영상 등)로 동기유발 및 관련성 찾기 활동으로 시작한다.

9주차는 본격적인 팀 프로젝트를 진행하기 전 미니 프로젝트를 진행하여 프로젝트에 대한 부담감을 줄이고 성공 경험을 하도록 구성한다. 이때 실생활 또는 전공(계열) 관련 문제를 제시하여 해결안을 도출한다.

10 ~ 14주차는 아이디어를 스스로 탐색·구상하여 구현 및 평가하며 팀원 및 팀 간의 의견공유와 피드백을 통해 창의적으로 산출물을 구현한다. 산출물 프로젝트 진행 단계를 수업 중에 적절한 시간을 배분하여 예시를 통한 확장 활동을 함께 진행한다(예: 산출물 기획 및 제작 2차시, 예시 1차시 또는 산출물 기획 및 제작 1차시, 예시 2차시로 활용). 또 프로젝트 계획서 및 보고서를 MCPS 수업 단계에 맞게 과제 제출하도록 한다.

마지막 15주차는 교내 일정 및 컴퓨팅 관련 행사 일정을 미리 확인하여 아이디어 확산 활동(교내 SW 전시회, 아이디어 경진대회, IoT 시연 등)을 진행하도록 구성하였다. MCPS 수업모형 기반의 컴퓨팅 수업 설계는 Table 2와 같이 나타내었다.

Table 2. Designing Computing Classes Based on MCPS Instructional Models

Weeks	MCPS Class Step	Class Topics and Teaching/Learning Activities
1	Step 1. Motivation and Preparation	Online survey to analyze learner characteristics - Understanding and my attitude towards the 4th Industrial Revolution - Recognition of the need for computing education - Activities related to major

Weeks	MCPS Class Step	Class Topics and Teaching/Learning Activities
2	Step 1. Motivation and Preparation	Understanding computing algorithms and computers, etc. (Writing an algorithm for eating lunch in a school cafeteria) - Creating and experiencing real-life algorithms
3	Step 2. Building of Basic Knowledge	Understanding Variables - Related IT issues - Real-life examples of using variables
4		Understanding conditional and loop statements - Related IT issues - Real-life examples of using conditional statements
5		Understanding loops and lists - Related IT issues - Real-life examples of using loops
6		Understanding lists and functions - Related IT issues - Real-life examples of using lists
7	Step 3. Application of Basic Knowledge	Other features (maps, web functions, sensors, etc.) and how to use them - Related IT issues - Real-life examples of other functions
8		Check understanding
9	Step 4. Planning Outputs and Execution	Solving problems related to daily life and major (field) - Presenting problems related to real life - Sharing and presenting solutions
10		Explore Ideas - Explore what interests you most - Bring out inner feelings and ideas - Deliverable Plan 1 : : describe freely
11		Ideation Express your ideas concretely - Related Activities, Deliverables Project - Deliverable Plan 2 : : Mind maps, brainstorming, sketching, etc.
12		Design and share ideas - Related Activities, Deliverables Project - Deliverable Plan 3 : : UI design, logical design, scene-specific design
13	Step 5. development	Idea implementation - Related Activities, Deliverables Project - Deliverable plan : 4 : : Contents to be developed and contents to be corrected
14		Output evaluation - Related Activities, Deliverables Project - Deliverable report : Project performance results and impressions
15	Step 5. development	Spreading ideas - Attending the on-campus software graduation exhibition - Participation in software performance presentations for non-majors

3.3 학습자 특성 분석

비전공자를 대상으로 MCPS 수업모형 기반의 컴퓨팅 수업을 적용하기 전 1차시 수업에서 학습자 특성 분석을 위

한 온라인 설문을 실시한 결과는 Table 3과 같다.

Table 3. Online Survey Results for Analysis of Learner Characteristics(N=19)

Survey Questions	Survey Content	Ratio
1. Experience using the program (Duplicable)	1) Not applicable (12 people)	63.16%
	2) Scratch (5 people)	26.32%
	3) Entry (4 people)	21.05%
	4) C language (3 people)	15.79%
	5) Python (2 people)	10.53%
	6) App Inventor (2 people)	10.53%
	7) Others (Java, PHP, etc.) (0 people)	0.00%
2. Team Project Proper Personnel	1) 4 people (10 people)	52.63%
	2) 3 people (3 people)	15.79%
	3) 5 or more people (3 people)	15.79%
	4) 2 people (2 people)	10.53%
	5) 1 person (1 person)	5.26%
3. How to select a team project	1) Selected by the instructor considering programming ability (8 people)	42.11%
	2) Choose your own team members (6 people)	31.58%
	3) Randomly selected (5 people)	26.32%
4. The biggest difficulty in computing classes?	① Psychological fears (fear of not being able to keep up, not understanding, etc.)	
	② Unfamiliarity with computers (typing speed, inexperience with computers, etc.)	
	③ Difficulty with the language (functions, difficulty with programming itself, etc.)	
	④ pace of class	
	⑤ slow understanding	
	⑥ Difficulty of class etc	
5. What do you want from class?	① Slow and easy explanation	
	② Classes that everyone can follow	
	③ Appropriate progression	
	④ Appropriate assignments	
	⑤ Appropriate duties,	
	⑥ Fair evaluation	
	⑦ Appropriate assignment etc	

학습자 특성 분석 결과를 살펴보면 프로그래밍을 위한 도구 사용의 경험 유무에서 해당없음(63.16%) 항목이 가장 높았으며 스크래치(26.32%), 엔트리(21.05%) 순으로 높게 나타났다. 앱인벤터(10.53%)는 대부분 처음 사용하는 것으로 나타났다. 팀 프로젝트의 적정인원은 4명(52.63%)으로 가장 높게 나타났으며, 팀 프로젝트 적정인원은 4명(52.63%)로 가장 높았으며 3명(15.79%), 5명 이상(15.790%), 2명(10.53%), 1명(5.26%) 순으로 나타났다. 팀 프로젝트 배정 방법은 교수가 정하기(42.11%)로 가장 높게 나타났으며 스스로 정하기(31.56%), 랜덤으로 정하기(25.32%)로 조사되었다. 또 학생들의 전공은 무역학과 다수를 포함한 사학과, 국어국문학과, 대부분 인문사회계열에 속하였다.

다음으로 개방형 응답을 살펴보면, 가장 어려운 점은 심리적 두려움(따라가지 못할 것 같아서, 이해하지 못할 것

같아서 걱정 등)이 가장 많았으며, 컴퓨터에 대한 미숙(타자속도, 컴맹 등), 언어의 어려움(프로그래밍 자체의 어려움, 함수 및 용어 자체의 어려움 등), 수업 속도, 수업의 난이도 등이 프로그래밍 수업의 가장 어려워하는 점으로 응답하였다. 또 본 수업에 바라는 점은 누구나 따라할 수 있는 수업을 희망하는 의견이 가장 많았으며 적절한 진행, 적절한 수업 속도, 적절한 난이도, 적절한 과제 등으로 조사되었다.

이에 본 수업의 2주차 부터 다음과 같이 학습자 특성 분석 결과를 적용하였다. 단계 1에서는 프로그래밍 수업의 경험이 부족한 학생들을 대상으로 기초 개념과 실생활 관련 예제로 필요성을 강조하여 실습 및 과제를 구성하였다(예 : 4차혁명과 무역학과의 연관성 및 나의 자세, 학식 먹기 알고리즘 작성하기 등).

단계 2에서는 인문사회계열 관련 뉴스나 최신 IT기술 등을 소개로 수업을 시작하도록 구성하였다. 1차시와 2차시는 개념을 이해하고 3차시는 생각하는 시간을 확보하기 위하여 1가지 기능을 자유롭게 추가하거나 변경하기 실습 및 과제를 하도록 구성하였다. 예를 들어 무역학과의 경우 무역학개론, 국제금융론, 국제상무론의 점수를 계산하여 평균과 평점을 구하거나 학점 계산하기 등의 실습 문제를 구성하였다. 또한 6주 ~ 7주차의 주제를 변경하였다. 가장 어려워하는 함수, 확장기능을 예제로 소개하는 것으로 대신하였고, 실생활에서 많이 사용되는 기타 기능(지도, 웹 기능, 바코드스캐너, 만보기 등)으로 변경하였다.

단계 3에서는 새로운 학습보다 실생활 예제의 문제를 일부 함께 해결하고 확장 기능 또는 변형하기 실습 및 과제를 구성하였다. 예를 들어 간단한 수도 이름 맞추기 퀴즈를 함께 작성하여 성공 경험을 하고 자유로운 확장 활동이나 과제로 점수 추가, 타이머 추가, 아이템 추가 또는 경계 및 무역 관련 퀴즈로 변경할 수 있도록 안내하였다.

단계 4에서는 학습자 의견을 반영하여 팀 프로젝트 인원을 4명으로 정하였으며, 팀원은 교수가 과제점수, 실생활 확장 예제 및 중간고사 실습 평가를 반영하여 정하였다. 12주차에 본교 소프트웨어 관련 전공학과의 졸업작품 전시회 일정이 있어 수업의 일부 시간을 할애하여 함께 관람한 후 소감문 작성하도록 하였다. 특히 앱인벤터 프로그래밍으로 개발된 작품들을 감상할 수 있어 좋은 기회를 제공할 수 있었다. 또 수업의 3차시에는 팀별 프로젝트 논의 시간을 부여함으로써 프로젝트 진행의 난점 진단 및 모니터링을 통해 구성원의 대인관계 조정, 원활한 진행과 결과물의 완성도를 높이는데 필요한 피드백 및 조력을 하였다.

단계 5에서는 본교 소프트웨어융합원에서 비전공자 기초교과목 대상으로 소프트웨어 성과물 전시회를 개최하게 되었다. 이에 본 수업의 최종 프로젝트 산출물을 비교과 과정 연계함으로써 보상 및 자신감을 고취할 수 있는 기회를 제공하였다.

따라서 본 수업을 수강하는 학습자의 특성은 프로그래밍 수업의 경험이 부족한 학습자들로 구성되었으며 심리적인

두려움이 가장 큰 어려움의 요인으로 분석되었다. 이를 위해 학습자 위주의 쉽고 흥미로운 주제로 다가가 중도 포기자가 발생하지 않도록 노력이 필요하며, 학습자에게 많은 선택권과 기회를 제공하여 성취감과 만족도를 높일 수 있도록 수업 프로그램을 구성하였다.

4. 연구 방법

4.1 연구 대상

본 연구는 경북권 소재 A 대학교 비전공자 대상 앱인벤터 프로그래밍 강좌를 수강하는 1학년 인문사회계열(무역학과, 사학과) 학생을 대상으로 연구를 수행하였다. 초기 연구 대상자 23명에서 휴학, 결석, 미참여 학생을 제외한 총인원은 19명으로 구성되었으며, 적용 기간은 2023년 9월 1일 ~ 12월 19일까지 15주간 실시하였다.

4.2 연구 설계

본 연구에서는 비전공자의 컴퓨팅 교양수업에서 MCPS 수업모형의 컴퓨팅 사고력 효과성을 검증하기 위하여 컴퓨팅 사고력 사전검사를 실시하였다. 사후검사는 15주 수업이 종료된 직후 실시하였다. 연구 설계는 Table 4와 같다.

Table 4. Research Design

G1	O1	X1	O2
----	----	----	----

G1 : Experimental group

O1 : Pre-test (Computational thinking tool)

X1 : MCPS-based computing class

O2 : Post-test (Computational thinking tool)

4.3 검사 도구

컴퓨팅 사고력 검사를 위한 표준화된 도구는 존재하지 않으나 이와 관련된 다양한 연구들이 진행되고 있다[25]. 본 연구를 위한 컴퓨팅 사고력 검사 도구로는 비버챌린지 문항을 활용하였으며, 단계별 평가를 위한 도구로 MCPS 수업모형 기반 평가루브릭을 사용하였다.

먼저, 비버챌린지(Bebras Challenge)는 정보과학에 관한 내용을 바탕으로 학습동기를 유발하고, 컴퓨팅 사고력을 함양하기 위한 목적으로 개발된 문제해결 기반의 평가 모델이다[32, 33]. 2004년 리투아니아에서 시작되어 2022년 기준 전 세계 61개국이 공식 회원국으로 참여하고 있다. 이러한 이유로 많은 연구자들이 컴퓨팅 사고력을 측정하기 위한 도구로 활용하고 있다. 또한 보편교육을 지향하며 컴퓨팅 사고력을 진단하고 학생들의 컴퓨팅 사고력을 촉진한다고 하였다[14, 32-34].

비버챌린지는 적용 연령층에 따라 6개의 그룹으로 출제되고 있다[32-34]. 본 연구에서는 컴퓨터 비전공 대학생의 수준을 고려하여, 비버챌린지의 연령 그룹 중 고등학생 연령대에 해당하는 6그룹의 문제를 선택하여 활용하였다. 비

버챌린지의 채점 방식은 문제의 난이도에 따라 부여된 점수가 다르고, 오답인 경우 역시 난이도에 따라 감점 처리가 다르다. 또한 과제당 약 3분 이내에 해결하고 그 결과를 바탕으로 컴퓨팅 사고력을 평가하는 방식으로 진행된다[32-34]. 본 연구의 문항의 난이도별 채점 방식은 Table 5와 같다.

Table 5. Evaluation Method According to Item Difficulty[25-28]

Difficulty Level	Correct answer	Incorrect	No response
Hard	+12	-4	0
Medium	+9	-3	0
Easy	+6	-2	0

비버챌린지의 평가 영역은 알고리즘과 프로그래밍(ALP), 자료/자료구조와 표현(DSR), 컴퓨터 처리와 하드웨어(CPH), 통신과 네트워킹(COM), 상호작용/시스템과 사회(ISS) 등 5가지가 있다[28-29]. 본 연구에서 사용한 문항은 컴퓨터과학의 알고리즘 및 프로그래밍(Algorithms and Programming, ALP) 영역으로 4지 선다형(5문항)과 주관식 문항(1문항)을 포함하여 총 6문항씩 구성하였다. 각 평가지 문항은 동일한 평가 영역의 동일한 난이도로 구성하였으며, 평가도구의 적합성과 타당성을 확보하기 위해 대학에서 컴퓨팅 교육 강의를 하는 박사학위 이상의 전문가 2인의 검토를 거쳤다. 본 연구에서 사용된 사전검사지 문항과 사후검사지 문항 및 난이도는 각각 Table 6, Table 7과 같다.

Table 6. Bebras Tasks of Pre-test Tool

No.	Task	Task-code (Country)	Difficulty	Age group
1	Truchet tiles	2021-AT-06 (Austria)	M	VI
2	Stickers	2020-FI-02 (Finland)	M	VI
3	Snow white	2021-IT-05 (Italy)	E	VI
4	Taking leaves	2021-KR-06 (South Korea)	M	VI
5	Ordering seven students	2021-ES-01 (Spain)	E	VI
6	Rabbit paddock	2020-CZ-04 (Czech republic)	E	VI

Table 7. Bebras Tasks of Post-test Tool

No.	Task	Task-code	Difficulty	Age group
1	Passwords	2020-DE-05b (Germany)	E	VI
2	Beaver sort	2021-DE-06 (Germany)	M	VI
3	A new neighbor	2020-KR-01 (South Korea)	M	VI

No.	Task	Task-code	Difficulty	Age group
4	compare	2021-LT-05 (Lithuania)	E	VI
5	Train trip	2020-SI-01 (Slovenia)	E	VI
6	Longest sequence	2021-UA-01a (Ukraine)	M	VI

다음으로 MCPS 평가 루브릭은 MCPS 수업모형 기반의 컴퓨팅 수업이 학습 목표에 맞게 교육이 이루어 지는지 평가하기 위한 도구로 개발되었다[17]. MCPS 수업모형 기반 평가루브릭은 13인의 전문가 검토를 통해 적절성이 확인되었으며, 프로그래밍 학습 과정과 창의적인 산출물 제작의 완성도 및 과정에 대하여 명확한 평가 준거를 제시하여 보다 객관적이고 신뢰성 있는 평가를 제공하는데 목적이 있다. 개발된 평가루브릭은 4가지 단계(능숙(5점), 발달(3점), 기초(1점), 무응답/미제출(0점))로 평가 수준을 나타내었다. 또 수업 단계별 평가 요소는 지식, 기능, 태도 영역을 포함하여 나타내었으며 자기평가 및 동료평가도 함께 구성하였다[17]. MCPS 수업모형 기반의 평가루브릭 평가 요소는 Table 8과 같다.

Table 8. Evaluation Rubric Evaluation Elements of the MCPS Instructional Model

MCPS Instructional Model Step	Evaluation Factors	Knowledge	Skills	Attitudes
Step 1. Motivation and Preparation	1.1 basic principles and concepts of computers	○		
	1.2 recognition of the relevance of computing to major fields	○		○
	1.3 design real-world computing algorithms	○	○	○
Step 2. Building of basic Knowledge	2.1 the concept of programming Elements	○		
	2.2 executing code with programming elements	○	○	
	2.3 executing code to solve real-world problems	○	○	○
Step 3. Application of basic Knowledge	3.1 analyzing code to solve real-world(major-related) problems	○		
	3.2 execute code to solve a real-world(major-related) problem	○	○	
	3.3 expressing a Solution	○	○	○

MCPS Instructional Model Step	Evaluation Factors	Knowledge	Skills	Attitudes
Step 4. Planning outputs and Execution	4.1 exploring real-world problems	○	○	
	4.2 appropriateness of data collection and analysis	○	○	
	4.3 decompose the problem into solvable units	○	○	
	4.4 design a solution	○	○	
	4.5 implement the solution	○	○	
	4.6 refine through testing and debugging	○	○	
	4.7 feedback through sharing and collaboration	○	○	○
	4.8 representing the results of collaborative output	○	○	○
Step 5. development	5.1 various expressions of ideas	○	○	○
	5.2 share your impressions of participating in computing classes	○	○	○

4.4 분석 방법

연구의 분석 도구로는 IBM SPSS Statistics 26 버전을 사용하여 분석을 실시하였다. 컴퓨팅 사고력의 변화를 알아보기 위해 독립표본 t-검정, 대응표본 t-검정을 실시하였으며 유의수준 $p < 0.05$ 로 분석하였다.

또한 혼합방법연구(mixed method research)를 통해 정량적 분석 결과와 컴퓨팅 교양수업 MCPS 평가루브릭 및 강의평가에 나타난 학생 의견을 정성적으로 분석한 결과를 함께 진술하였다. 정성적 자료 분석은 프로젝트 수행 소감과 강의평가 의견의 키워드를 분석하여 워드 클라우드(Word cloud)와 토픽 모델링(Topic Modeling) 기법으로 분석하였다. 한글 텍스트 분석을 위해서는 국내에서 개발한 오픈소스 프로젝트인 KoNLPy 패키지를 사용하여 키워드의 빈도수를 계산하였으며, 토픽 분석을 위해서는 파이썬의 gensim 패키지에서 제공하는 LDA(Latent Dirichlet Allocation) 토픽 모델링을 사용하였다[35].

5. 연구 결과

5.1 수업의 효과 : 정량적 분석

본 연구에서는 비전공자를 대상으로 MCPS 수업모형 기반의 컴퓨팅 교양수업을 진행하였고 수업의 효과성을 살펴보기 위해 3가지 측면(컴퓨팅 사고력 결과, MCPS 수업모형 기반의 컴퓨팅 교양수업 성취도 평가 결과, 강의평가 결과)에서 정량적 분석을 실시하였다.

첫째, 컴퓨팅 사고력 결과분석을 위해 평균과 표준편차, 성별에 따른 기술통계 및 평균 비교를 통한 성취도 분석을 진행하였다. 사전, 사후 컴퓨팅 사고력 검사 결과를 바탕으로 대응표본 t-검정을 실시한 결과는 Table 9와 같다.

Table 9. Computational Thinking Ability(ALP) Test Results

Factor	Pre-test (N=19)		Post-test (N=19)		t	p
	M	SD	M	SD		
Computational Thinking Ability (ALP)	0.68	11.85	6.47	8.80	-2.196	.041*

컴퓨팅 사고력 사전, 사후검사 결과를 살펴보면 사전 검사의 평균은 0.68(SD: 11.85)이고 사후검사의 평균은 6.47(SD: 8.80)로 나타나 사전검사에 비해 높은 상승이 관찰되었다. 또 p값은 .041로 유의 수준 .05 미만으로 나타나 통계적으로 유의한 차이를 보였다.

이를 통해 비전공자의 대학 컴퓨팅 교양수업에서 MCPS 수업모형 기반의 컴퓨팅 수업은 컴퓨팅 사고력을 향상하는데 있어 유의미한 효과가 있음을 확인하였다. 또한 성별에 따라 차이가 있는지 확인하기 위해 사전검사와 사후검사의 컴퓨팅 사고력 점수를 바탕으로 평균과 표준편차를 분석하였다. 독립표본 t-검정을 실시한 컴퓨팅 사고력 점수의 사전검사와 사후검사의 결과는 각각 Table 10, Table 11과 같다.

Table 10. Computational Thinking Ability Pre-test Results According to Gender

Division	N	M	SD	t	p
Male	13	2.15	12.542	.787	.442
Female	6	-2.50	10.502		

사전검사 결과를 살펴보면, 남학생의 평균점수는 2.15(SD : 12.542)이며, 여학생의 평균점수는 -2.50(SD : 10.502)로 나타났다. 성별 따른 평균의 차이에 대한 유의확률 p값이 .442로 나타나 성별에 따라 차이가 없는 것으로 나타나 두 집단이 동질 집단을 확인하였다.

Table 11. Computational Thinking Ability Post-test Results According to Gender

Division	N	M	SD	t	p
Male	13	8.08	8.808	1.181	.254
Female	6	3.00	8.462		

사후검사 결과를 비교해 보면, 남학생의 사후검사 평균점수는 8.08(SD : 8.808)이며 여학생의 평균점수는 3.00(SD : 8.462)로 나타났다. 성별 따른 평균의 차이에 대한 유의확률 p값이 .254로 나타나 그 차이는 통계적으로 유의미하지 않았다. 이를 통해 비전공자의 컴퓨팅 교양수업에서 MCPS 수업모형 기반의 컴퓨팅 수업을 통한 컴퓨팅 사고력 성취도의 향상 정도는 성별에 따라 통계적으로 유의미한 차이가 없음을 확인하였다.

따라서 MCPS 수업모형 기반의 컴퓨팅 교양수업은 컴퓨팅 사고력에 유의미한 차이가 있었고 성별에 따라 유의미

한 차이는 없었다.

둘째, MCPS 수업모형 기반의 컴퓨팅 수업에서 평가루브릭을 바탕으로 실시한 수업 전 과정에서의 성취도 평가 결과를 살펴보았다. 평가루브릭을 통한 최종 성취도는 수업에 참여한 실험집단 19명의 평균 점수(5점 만점)로 제시되었다. MCPS 수업단계별 컴퓨팅 교양수업 성취도 평균을 분석한 결과는 Table 12와 같다.

Table 12. Evaluation rubric results of the score experimental group

MCPS Instructional Model Steps	Evaluation Type	Evaluation Factors	Evaluation Average (n=19)
step 1. Motivation and preparation	Midterm exam (written)	1 basic principles and concepts of computers	4.13
	Assignments & Feedback	1.2 recognition of the relevance of computing to major fields	4.53
	Training	1.3 design real-world computing algorithms	4.06
step 2. Building of basic knowledge	Midterm exam (Written + Practical)	2.1 the concept of programming Elements	4.68
		2.2 executing code with programming elements	4.21
		2.3 executing code to solve real-world problems	3.78
step 3. Application of basic knowledge	Practice + Assignment	3.1 analyzing code to solve real-world(major-related) problems	3.91
		3.2 execute code to solve a real-world(major-related) problem	
		3.3 expressing a Solution	3.85
step 4. Planning outputs and execution	Project Evaluation (plan, report)	4.1 exploring real-world problems	4.84
		4.2 appropriateness of data collection and analysis	
		4.3 decompose the problem into solvable units	4.85
		4.4 design a solution	
		4.5 implement the solution	3.92
		4.6 refine through testing and debugging	
4.7 feedback through sharing and collaboration	4.35		
4.8 representing the results of collaborative output	4.49		
step 5. Development	Assignment	5.1 various expressions of ideas	4.37
overall average			4.30

MCPS 평가루브릭을 바탕으로 분석한 결과 전체 평균은 4.30점으로 '능숙'에 가까운 결과가 나타났다. 또한 수업 단계별 평가 결과 중 낮은 성취도를 보인 부분을 살펴보면, 단계 2의 '2.3 실생활 문제의 해결안을 위한 코드 실행(3.78점)'으로 가장 점수가 낮았으며 다음으로는 단계 3의

‘3.3 문제해결안 표현(3.85점)’, 단계 4의 ‘4.5 문제해결안 구현과 및 4.6 테스트와 디버깅을 통한 수정(3.92점)’ 순으로 평균 점수가 낮게 나타났다. 이는 학생들이 프로그래밍 개념과 기본요소는 이해하지만 이를 소프트웨어로 표현하거나 활용하는 부분을 다소 어렵게 느끼는 것으로 해석할 수 있다.

수업 단계 4에서는 산출물 제작 과정 및 산출물 결과를 다양하게 평가하기 위해 학생들이 제출한 계획서, 보고서, 온라인 폼을 이용한 교수평가, 동료평가(팀내, 팀간) 및 자기평가를 포함하였다. 평가 내용은 아이디어 탐색 및 분석(20%), 아이디어 구상(20%), 설계(20%), 구현 및 평가(20%), 기타(기능성, 창의성, 사용성(20%)) 등의 항목을 포함하여 다양한 각도로 구성하여 평가를 실시하였다. 마지막 항목으로 좋은점과 개선점에 대한 개방형 의견을 작성하여 평가가 종료된 직후 피드백을 공개하였다. 이를 통해 수업 단계 4의 전체 평균 점수가 다른 수업 단계보다 높게 산출된 것을 확인할 수 있는데, 이는 수업 단계 4에서 팀 내 또는 팀 간의 공유와 협력 그리고 피드백을 통해 수업의 성취도에 긍정적인 영향을 미친 것으로 해석할 수 있다.

셋째, 컴퓨팅 수업을 적용한 후 시행된 학생들의 강의평가 결과(만족도)를 살펴보았다. 전체 교양수업과 컴퓨팅 교양수업 강의평가 결과는 Table 13와 같다.

Table 13. Computing Liberal Studies Course Evaluation Results

Course Evaluation Items	Rating Average	
	Experimental group	All Liberal Arts Class
1. Degree of achievement of the learner's class goals	4.50	4.31
2. Degree of help of class content and class materials	4.67	4.38
3. Degree of consistency and systematic progress between actual classes and classes plans	4.67	4.42
4. Instructor's class preparation and teaching ability	4.67	4.41
5. Utilize teaching methods appropriate for class goals	4.75	4.42
6. Communication between instructor and learner	4.67	4.40
7. Appropriateness and interest of the task	4.75	4.36
8. Helps improve core competencies	4.67	4.37
overall average	4.67	4.38

전반적인 수업 만족도는 모든 항목에서 전체 강좌 평균보다 높게 나타났다. 특히 ‘수업 목표에 적절한 교수법 활용’ 항목과 ‘과제의 적절성과 흥미도’ 항목은 4.7 이상으로 매우 높게 조사되었다. 또 실험집단의 평균은 4.67로 나타나 교양교육원 전체 교양교과목 평균에 비해 높게 나타났다.

본 수업의 효과성을 보다 면밀하게 확인하기 위해서는 동일 교과목의 세부항목을 비교하는 것이 타당하지만, 동

일 교과목의 세부항목은 공개되지 않고 동일 교과목 전체 평균만 공개되어 있어 수업의 효과성을 확인하였다. 즉 동일 교과목(앱인벤터)에 해당하는 7개 분반의 수업평가 평균은 4.15로 나타났으며 실험집단의 평균은 4.67로 나타나 학습자의 만족도가 전반적으로 높았다고 할 수 있다.

따라서 비전공 학생들의 컴퓨팅 사고력 검사, MCPS 평가루브릭을 통한 컴퓨팅 수업 성취도 평균 점수를 분석한 결과 비전공 학생들의 MCPS 수업모형 기반의 컴퓨팅 수업에서 컴퓨팅 사고력 향상에 긍정적인 변화가 있었음을 관찰할 수 있었다. 또 강의평가 결과를 통해 살펴본 MCPS 수업모형 기반의 컴퓨팅 교양수업은 수업 목표에 맞는 적절한 교수법과 학생의 수준과 흥미를 고려한 과제가 중요하다는 것을 유추할 수 있었다.

5.2 수업의 효과 : 정성적 분석

앞서 정량적 분석을 통해 MCPS 수업모형 기반의 컴퓨팅 수업은 컴퓨팅 사고력 향상에 긍정적인 변화를 확인할 수 있었다. 그러나 정량적으로 나타나지 않은 부분을 확인하기 위해 MCPS 단계별로 수업에 사용된 과제, 실습 예제, 프로젝트 등에 대하여 학생들의 반응 위주로 정성적 분석을 통해 살펴보았다. 평가 및 단계별 변화를 확인하기 위해 매 수업 종료 직후 또는 다음 수업 전까지 LMS에 제출된 자료, 프로젝트 계획서와 보고서 및 학생 관찰을 중심으로 분석하였다.

먼저, 단계 1의 1주차에 4차 산업혁명에 대해 알아본 후 과제로 부여된 ‘4차 혁명과 전공과의 연관성 및 나의 자세’ 관한 학생과제물에 대한 예시이다.

학생 A : 4차 산업 인문학은 많은 고고학자들도 읽지 못했던 고대 석판을 AI가 해석을 하는 사례가 늘어나면서 고고학 분야에서도 인공지능을 통해 많은 발전을 나타내고 있다. [일부생략...] 이렇게 4차 산업 발달이 인문학에 있어 기존과는 다른 새로운 방안을 나타내고 있다는걸 알게 되었다(인문계열).

학생 B : 4차 산업혁명은 준비된 사람에게는 기회고 그렇지 않은 사람에게는 위기일 것이다. 사회는 공평하게 기회를 제공한다고 하지만 기술이 발전될수록 돈이 많고 더 많이 배운 사람이 더 많은 혜택을 누릴 가능성이 높아진다. [일부생략...] 교수가 꿈인 나의 진로에 있어서 교육만 하는 역할은 사라질 수 있다. 따라서 도태되지 않기 위해 교육 외 더 다양한 분야의 공부와 최소한의 혜택을 누릴 재산을 모으는 것이 나의 목표이다(사회계열).

이외에도 정보의 편리성, 생활의 편의성 등 긍정적인 의견과 생계를 위협하는 것, 사생활 침해, 인력이 기계로 대체되는 등 부정적이라는 의견도 있었으며 수업을 계기로 미래를 대비하는 자세와 마음가짐을 확인할 수 있었다고 응답하였다. 이에 대한 교수자의 피드백의 일부를 살펴보면 다음과 같다.

학생 A 피드백 : 4차 산업혁명이 인문학, 특히 고고학에 미치는 영향을 잘 설명하고 있네요. AI가 고대 유물 해석에 기여하는 사례를 통해 기술과 인문학의 융합을 강조한 점이 인상적입니다. 앞으로의 연구와 학습에서도 이러한 학제간의 접근을 지속적으로 탐구하길 바라며 좋은 통찰력을 보여주었습니다.

학생 B 피드백 : 4차 산업혁명이 개인의 준비 정도에 따라 기회와 위기로 나뉜다는 점을 잘 짚어냈습니다. 기술 발전이 불평등을 심화할 수 있다는 우려는 매우 중요한 문제입니다. 교수로서의 역할 변화에 대한 인식도 인상적이며, 다양한 분야의 학습과 재산 축적을 목표로 하는 자세가 필요합니다. 앞으로의 교육 방향성과 개인의 발전을 위해 지속적인 자기계발이 중요하다는 점을 강조하고 싶네요.

다음으로 단계 2의 3주차에는 변수의 개념을 설명하고 기본적인 산술연산(더하기, 빼기)을 따라하면서 코딩한 후 학습자가 스스로 곱하기, 나누기를 코딩하도록 하였다. 이 과정에서 간단한 프로그램을 함께 작성한 후 학습자 모두가 직접 실행 결과화면을 확인하고 나서 다음 단계로 진행하였다. 이후 확장 활동으로 계산기를 활용하여 나만의 계산 기능을 추가하여 예제를 완성하도록 부여하였다. 수업 종료 직전 내용을 발표하게 하였고 미완성은 과제로 제시하였다. 그 결과 대부분 교수자가 사각형의 넓이를 구하는 예시를 참고하여 원의 넓이 구하기, 삼각형의 넓이 구하기, 사다리꼴의 넓이 구하기에 집중이 되었다. 그러나 수면시간 구하기, 더치페이 계산하기, 등록금을 내기 위해 매달 저축해야 할 금액 계산하기, 교통비 계산하기, 점심값 계산하기 등 일상생활의 문제를 해결하는 기능을 추가한 예제를 볼 수 있었다. 간단한 앱을 통해 처음으로 SW를 이용하여 실행되는 화면을 보며 신기하고 뿌듯해하는 학습자들의 모습을 확인할 수 있었다.

다음은 단계 3의 9주차에서 간단한 수도 이름 맞추기 퀴즈 앱을 함께 작성하여 성공적인 경험과 함께 리스트를 이해하도록 하였다. 문제기반 학습으로 학습자가 흥미 있는 주제를 찾아서 퀴즈를 출제하고 확장 기능을 1개 이상 추가하여 코딩하도록 하였다. 실습 결과 예제와 다른 나라와 수도명으로 변경하는 몇몇 학생들이 있었지만 넌센스 퀴즈, 영어 단어 퀴즈, 동물 이름 맞추기, 전공 관련 문제, 아이돌 이름 맞추기 등 다양한 주제로 나타났다. 확장 기능으로는 맞춘 개수, 틀린 개수, 맞추거나 틀렸을 때 소리 추가, 점수 추가, 힌트 추가, 이미지 추가 등 앞에서 학습했던 텍스트 조건문이나 미디어 기능을 활용하여 나타내었다. 반면 학습하지 않았던 타이머 기능, 알림 기능들도 스스로 찾아보면서 해결하거나 교수자에게 질의를 하는 모습을 관찰할 수 있었다. 이를 통해 학생들이 관심있어하는 부분을 퀴즈로 표현함으로써 학습의 부담감을 줄이고 앞에서 학습했던 기능(변수, 제어문, 반복문, 미디어 등)을 이용해 스스로 문제를 해결하는 모습을 볼 수 있었다.

다음은 단계 5의 12주차에 진행되었던 소프트웨어 관련

전공학과 졸업작품 전시회를 관람한 후 과제로 관람 소감문을 제출하게 하였다. 강의 계획서에는 15주차로 계획되었지만 학내 행사 일정상 변경하여 진행하였다. 다음은 전시회 관람 소감문의 일부이다.

학생 C : ‘우리가 수업했던 앱인벤터를 이용한 전시가 있어서 놀랐고 어떻게 이렇게까지 할 수 있는지 대단하다는 생각이 많이 들었다. 앱인벤터를 처음 접했을 때는 그냥 교수님이 하시는 거 보고 마냥 따라만 해서 혼자 하라고 하면 어떻게 하는지 자세히 알지 못했는데 이번 SW전시회를 보고 나서부터는 혼자 스스로 도전해보고 전시회만큼은 아니지만 스스로 할 수 있게 노력을 해봐야겠다고 다짐하게 되는 시간이었다’

학생 D : ‘수업이 끝난 이후 SW전시회를 갔는데 대구에 있는 명소를 소개해주는 앱을 발견했다. 고향인 대구를 여기서 보니 공감 가는것도 많고 인상 깊었다. 나도 다음에 이런 훌륭한 어플을 만들고 싶다는 생각이 들었다’

이외 폐기물 처리 앱, 지진 감지 프로그램, 인터랙티브 웹, 비 사이로 막가 게임, 젤리 키우기 게임 앱 등 이런 작품들을 대학생이 만들 수 있다는 점과 창의성이 돋보이는 작품들이 인상 깊었다는 등의 의견을 볼 수 있었다.

단계 4의 12주 ~ 14주에서 프로젝트 진행과정, 수행소감 및 수업평가 개방형 응답의 출현 빈도가 높은 키워드를 분석하였다. 이를 위해 형태소의 뜻과 문맥을 고려하여 품사를 붙이는 품사 태깅 패키지를 사용하여 단어별 출현횟수를 계산하여 2이상인 키워드만 저장하여 Figure 1과 같이 나타내었다.

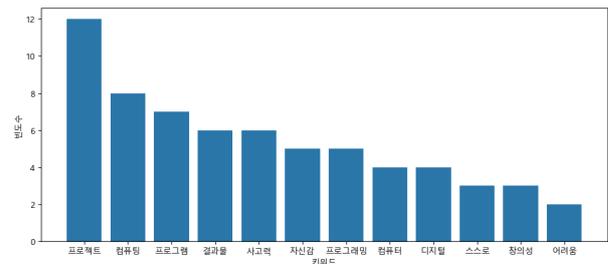


Figure 1. Keyword : Results Visualization

가장 많이 출현한 단어는 ‘프로젝트’, ‘컴퓨팅’, ‘프로그램’, ‘결과물’, ‘사고’, ‘자신감’, ‘프로그래밍’, ‘컴퓨터’, ‘디지털’, ‘스스로’, ‘창의성’, ‘어려움’으로 나타났다. 이러한 키워드는 프로젝트 기반 프로그래밍 수업에서 긍정적인 경험을 잘 나타낸다고 해석할 수 있다.

생성된 단어별 키워드는 빈도수를 계산하여 알아보기 쉽게 시각화하여 나타내었다. 워드 클라우드 생성 결과는 Figure 2와 같다.



Figure 2. Word cloud : Results visualization

또한 개방형 응답의 주제를 도출하기 위해 추출할 토픽의 개수(k=4)로 설정하였으며 토픽을 구성하는 주요 단어(words=15)에 따라 토픽에 대한 영향력 비율과 함께 출력된 것을 확인 할 수 있다. Table 14은 토픽 분석 결과를 나타낸다.

Table 14. Topic analysis results

Topic no.	Keyword(15)	Topic Label
1	0.024*코딩 + 0.023*처음 + 0.021*수업 + 0.021*프로젝트 + 0.019*자신감 + 0.017*인벤 + 0.014*생각 + 0.014*결과물 + 0.013*성취 + 0.013*흥미 + 0.013*문제 + 0.012*해결 + 0.012*기회 + 0.012*역량 + 0.012*향상	흥미 위주 문제 해결로 성취 및 자신감 향상
2	0.024*코딩 + 0.021*수업 + 0.020*프로그램 + 0.019*직접 + 0.017*생각 + 0.016*프로그래밍 + 0.015*인벤 + 0.015*프로젝트 + 0.015*진행 + 0.015*흥미 + 0.014*컴퓨팅 + 0.011*사용 + 0.011*통해 + 0.011*평소 + 0.011*관심	평소 생각을 직접 프로그래밍
3	0.052*수업 + 0.036*코딩 + 0.028*컴퓨팅 + 0.026*사고력 + 0.024*프로젝트 + 0.021*생각 + 0.020*인벤 + 0.016*기회 + 0.015*해결 + 0.013*결과물 + 0.012*프로그램 + 0.012*흥미 + 0.012*디지털 + 0.012*통해 + 0.012*향상	컴퓨팅 사고력 및 디지털 능력 향상
4	0.029*수업 + 0.023*코딩 + 0.022*프로젝트 + 0.018*통해 + 0.018*생각 + 0.014*성공 + 0.014*실력 + 0.013*창의성 + 0.012*우리 + 0.012*흥미 + 0.012*대로 + 0.012*항상 + 0.011*협동 + 0.010*인벤 + 0.010*처음	코딩을 통한 창의성

토픽 분석 결과를 살펴보면 프로젝트 기반의 코딩 수업이 '코딩', '수업', '프로젝트' 단어가 가장 높은 비율로 나왔다. 토픽에 대한 레이블은 자동으로 만들어지지 않기 때문에 프로젝트 기반의 코딩 수업에 해당하는 단어를 제외한 키워드를 고려하여 토픽 내용을 설명하는 레이블로 결정하였다.

다음은 LDA 토픽 분석의 결과를 시각화하기 위해 겹치지 않게 클러스터링하여 출력하였다. 화면의 왼쪽에는 토

픽 간 거리 지도가 있고 오른쪽 영역에는 토픽에서 관련성 높은 30개 단어를 나타내는 바차트로 시각화 하였다[31]. Figure 3 ~ Figure 6은 LDA 토픽 분석 결과를 시각화하여 나타낸 것이다.

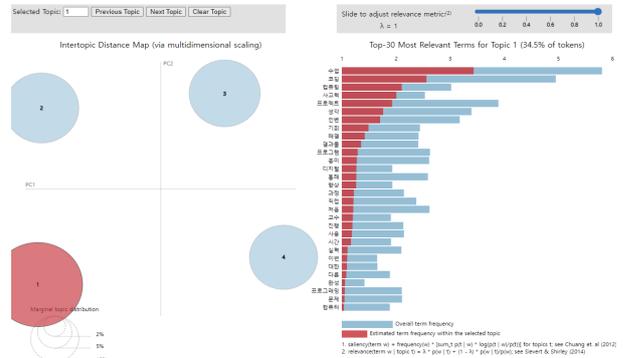


Figure 3. LDA topic analysis results : Topic1 Visualization

Figure 3과 같이 토픽 1은 '처음', '자신감', '생각', '결과물', '성취', '흥미', '문제', '해결', '기회', '역량', '향상' 등으로 나타났다. 토픽레이블은 '흥미 위주 문제해결로 성취 및 자신감 향상'과 관련된 의견이 있었다. 주요 단어를 고려하여 관련 학생 소감을 살펴보면 다음과 같다.

학생 E : 처음에는 힘들었지만, 직접 결과물을 코딩하고 해결하니 더욱 흥미가 생기고 성취감과 자신감이 생겼다.

학생 F : 프로그래밍 언어를 처음 접하게 되었고, 처음으로 앱인벤터를 사용하게 되었지만, 교수님과 팀원의 도움으로 흥미가 생겼고 일상생활 문제를 잘 해결할 수 있었다.

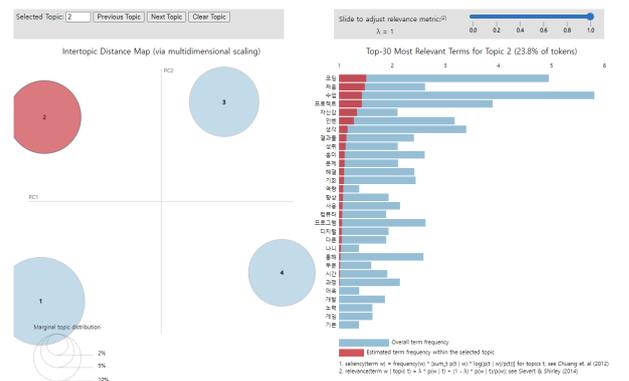


Figure 4. LDA topic analysis results : Topic 2 Visualization

Figure 4와 같이 토픽 2는 '프로그램', '직접', '생각', '프로그래밍', '진행', '사용', '평소', '관심' 등으로 나타났다. 토픽레이블은 '평소 생각을 직접 프로그래밍'하는 내용과 관련된 의견이 있었다. 이와 관련한 학생 소감은 다음과 같다.

학생 G : 직접 아이디어를 내고 앱 프로그램을 개발하고 사용해 보니 많은 과정이 필요하다는 것을 알게 되었다.

학생 H : 평소 생각할 시간을 많이 주시고 스스로 해결

로젝트 산출물을 계획하고 완성하는 과정에서 ‘흥미와 성취감 및 자신감 향상’, ‘평소 생각을 직접 프로그래밍’, ‘컴퓨팅 사고력 및 디지털 능력 향상’, ‘코딩을 통한 창의성’ 등의 토픽레이블이 관찰되었다. 이러한 변화는 인지적인 측면 뿐만 아니라 정의적인 변화에도 영향을 미쳤다는 것을 알 수 있었다. 이처럼 MCPS 수업모형 기반의 컴퓨팅 수업은 비전공자의 컴퓨팅 사고력의 긍정적인 변화를 유도할 수 있었음을 확인하였다.

6. 결론 및 제언

6.1 결론

본 연구는 비전공자의 컴퓨팅 사고력 향상을 위해 MCPS 수업모형에 기반한 컴퓨팅 수업을 설계하여 그 효과성을 분석하였다. 이를 위해 비전공 학습자의 특성을 진단하고 MCPS 수업모형 기반의 컴퓨팅 교양수업을 설계하여 현장에 적용하였다. 컴퓨팅 사고력 변화를 확인하기 위해 비버챌린지 문항과 MCPS 수업모형 평가루브릭을 검사지로 활용하였으며 수업 실시 전과 후의 검사 결과를 분석하였다. 본 연구의 결과를 바탕으로 결론을 살펴보면 다음과 같다.

첫째, 정량적 분석 결과 MCPS 수업모형 기반의 컴퓨팅 교양수업은 비전공자의 컴퓨팅 사고력 향상에 긍정적인 영향을 미치는 것을 확인하였다. 성별에 따라서는 통계적으로 유의한 차이가 관찰되지 않았다. 이러한 결과는 학습동기 또는 창의적 문제해결력을 위한 컴퓨팅 수업이 학생들의 컴퓨팅 사고력을 향상시키는 기존의 선행연구와 맥락을 같이 한다[6, 7, 17, 29, 30, 36, 37]. 즉 학습자의 특성을 고려한 흥미로운 수업 설계와 학습자 스스로 문제를 찾아 컴퓨팅 사고력을 통해 해결하는 과정 그리고 도전 의식과 성취감의 기회를 제공하도록 환경을 구성하는 것은 컴퓨팅 사고력 함양을 위해 필수적으로 요구되어야 할 것이다.

둘째, MCPS 수업모형 기반의 컴퓨팅 교양수업 성취도 평가 결과, 수업의 단계가 진행될수록 학생들의 성취도가 긍정적으로 향상되는 경향을 확인할 수 있었다. 즉 비전공자의 특성상 프로그래밍 수업 및 프로젝트에 대한 부담감이 존재할 수 있으며 이를 극복하기 위해서는 단계별로 스스로 성장하고 발전할 수 있도록 예시를 점차 줄여가며 공유와 협력 그리고 피드백의 필요성이 강조된다.

셋째, 수업이 종료된 직후 시행된 수업 평가 결과에서는 ‘수업 목표에 적절한 교수법 활용’과 ‘과제의 적절성과 흥미도’의 항목이 매우 높게 나타난 것을 확인할 수 있었다. 이는 MCPS 수업모형의 단계별로 교수법을 다양하게 적용하는데 중점을 두었기 때문으로 해석된다. 이러한 결과는 비전공 학생들이 수업에 대한 흥미를 느끼고 효과적으로 학습 목표를 달성할 수 있도록 중요한 요소로 작용한다[7, 11]. 따라서 비전공 학생들의 중도 포기를 예방하고 주제에 맞는 다양한 교수법을 활용하여 이해하기 쉽고 해

결하기 적절한 과제와 학습으로 구성하여야 할 것이다.

넷째, 정성적 분석을 통해 토픽 레이블을 살펴본 결과 4가지 토픽 ‘흥미와 성취감 및 자신감 향상’, ‘생각을 직접 프로그래밍’, ‘컴퓨팅 사고력 및 디지털 능력 향상’ 그리고 ‘코딩을 통한 창의성’ 토픽을 관찰할 수 있었다. MCPS 초기 단계에서 처음 실습 결과화면을 확인할 때 모든 학습자가 결과화면을 확인하여 성공 경험을 느낀 후 다음 차시로 진행하는 것이 중요하다. 즉 초반의 성공 경험은 학습자의 두려움을 감소시켜 학습이 보다 쉽게 느껴지도록 하여 도전해볼 만하다는 인식을 심어 준다. 이러한 경험은 한 학기 동안 포기하지 않고 끝까지 참여할 수 있는 지속동기가 되었음을 알 수 있었다.

이러한 연구 결과는 기존 연구와 비교했을 때 MCPS 기반의 수업이 실습이나 문법 중심의 교육방식에서 벗어나 [6-8] 기초지식 형성부터 응용 프로젝트 산출물 제작 과정까지 포괄적으로 학습할 수 있도록 유동적으로 설계되었다. 뿐만 아니라 컴퓨팅 수업에서 정서적 지원의 중요성을 강조하는 기존 연구와 일맥 상통한다[17, 21, 38-40]. 비전공자들이 컴퓨팅 수업의 두려움을 극복하고 자신감을 가지고 학습에 참여할 수 있도록 정서적 지원을 MCPS 수업모형을 통해 실현함으로써 컴퓨팅 교육에 대한 긍정적인 경험을 쌓을 수 있도록 함으로써 상당히 의미가 있다. 또한 지금까지 비전공자 대상으로 개발된 수업모형은 컴퓨팅 사고력 신장을 위해 컴퓨팅 사고력의 스킬 요소들을 학습 절차에 반영하여 이를 향상시키는데 목적이 있는 반면[27, 28, 41, 42]. 본 연구는 학습동기와 창의적 문제해결력을 함께 고려한 MCPS 수업모형을 이용하여 비전공 학습자의 어려움을 해결하기 위한 다양한 기법들을 단계별로 적용하여 설계 현장 적용 가능성을 높이는 실용적 시사점을 제공한다는 데 의의가 있다.

6.2 한계점 및 제언

본 연구는 비전공자를 대상으로 MCPS 수업모형의 효과성을 분석하였으나 다음과 같은 한계점이 있다. 첫째, 연구는 경상북도 소재 A 대학교의 1학년 19명에 대해 한 학기 동안 진행된 수업을 기반으로 하였기 때문에, 연구 결과의 일반화 가능성에 제한이 있다. 특정 대학과 학년의 학생들만을 대상으로 하였기에 다양한 전공 및 학년의 학생들을 포함한 보다 폭넓은 샘플이 필요하다. 향후 연구에서는 다양한 대학과 지역에서 비전공 학생들을 대상으로 한 연구를 진행하여 샘플 크기를 확대하고 연구 결과의 일반화 가능성을 높이는 연구가 진행되어야 할 것이다.

둘째, 본 연구는 단일 집단에 대한 효과성을 검증하였으므로, MCPS 수업모형을 적용하지 않은 통제 집단과의 비교를 통해 더욱 명확하게 입증하기 위한 추가적인 연구가 필요하다. 추후 연구에서는 MCPS 수업모형을 적용한 실험 집단과 적용하지 않은 통제집단을 설정하여 두 집단 간의 성과를 비교하는 연구를 수행해야 한다. 이를 통해 MCPS 수업모형의 효과성을보다 명확하게 입증할 수 있다.

셋째, 연구 기간이 한 한기로 제한되어 있어, 장기적인 효과를 평가하기에는 부족하다. 컴퓨팅 사고력은 단기간에 형성될 수 있는 능력이 아니므로 지속적인 학습과 경험이 필요하다. 향후 연구에는 장기적인 관찰 연구를 통해 MCPS 수업모형의 지속적인 효과를 평가할 필요가 있다.

또한 컴퓨팅 사고력 함양에 있어 성별에 따른 변인 외에 학업 성취도, 컴퓨팅 관련 경험, 학습동기 등 다양한 변인을 함께 고려한 추가연구를 진행한다면 MCPS 수업모형의 효과성을 강화할 수 있을 것이다.

본 연구를 통해 향후 컴퓨팅 사고력을 함양하기 위한 대학의 비전공자 컴퓨팅 교양 교육의 기초 자료로 활용될 것 기대한다.

참고문헌

- [1] Joint Ministries of South Korea. (2022, September 15). *Comprehensive Plan for Digital Talent Development*. <https://www.moe.go.kr/boardCnts/viewRenew.do?boardID=72769&boardSeq=92573&lev=0&searchType=null&statusYN=W&page=1&s=moe&m=0315&opType=N>
- [2] Ministry of Education. (2022, December 22). *2022 Revised Elementary, Secondary, and Special Education Curriculum Finalized and Announced*. <https://www.moe.go.kr/boardCnts/viewRenew.do?boardID=294&lev=0&statusYN=W&s=moe&m=020402&opType=N&boardSeq=93459>
- [3] Kim, J., & Sohn, E. (2021). Development of Data Science Course and Analysis of Computational Thinking Effect for non-Major Students. *The Journal of Korean association of computer education*, 24(3), 23-31. <https://doi.org/10.32431/kace.2021.24.3.003>
- [4] Korea National Institute for General Education. (2022, December 30). *2022 Standard Model for University Liberal Arts Basic Education*. https://www.konige.kr/data/general_edu.php
- [5] Korea National Institute for General Education. (2024, June 24). *2024 Liberal Arts Education Recommendations for Community Colleges*. https://www.konige.kr/notice/notice_view.php?bbs_cd=2024042400001&stx=&bsearch=c_title&pageno=1
- [6] Noh, J. (2023). Analysis of the Effectiveness of Liberal SW Education focused on Developing Computational Thinking and Creative Problem Solving Ability. *Journal of Industrial Convergence*, 21(1), 123-135. <https://doi.org/10.22678/JIC.2023.21.1.123>
- [7] Shin, Y. (2022). *An empirical case study on the problems and improvement direction of coding and software education : Focusing on the comparison between majors and non-majors in H University*. [Master's thesis], Hoseo University
- [8] Park, G., & Choi, Y. (2018). Exploratory study on the direction of software education for the non-major undergraduate students. *Journal of Education & Culture*, 24(4), 273-292. <https://doi.org/10.24159/joec.2018.24.4.273>
- [9] Pi, S. (2020). Learning Effects of Flipped Learning based on Learning Analytics in SW Coding Education. *Journal of Digital Convergence*, 18(11), 19-29. <https://doi.org/10.14400/JDC.2020.18.11.019>
- [10] Koh, E. (2021). *Development of a Python programming learning support system based on computational thinking* [Doctoral dissertation], Ewha Womans University.
- [11] Kang, E., Shin, S., & Lee, K. (2019). Education Model Using PBL for IT Convergence Education of Non-Major in Liberal Arts Class: Focusing on Computing Thinking. *Journal of Digital Contents Society*, 20(11), 2159-2168. <https://doi.org/10.9728/dcs.2019.20.11.2159>
- [12] Nah, J. (2017). Software education needs analysis in liberal arts. *Korean Journal of General Education*, 11(3), 63-89
- [13] Kim, H. (2018). A Study on the Design and Operation of Liberal Arts College for Improving Creativity of University Students. *The Journal of Creativity Education*, 18(4), 91-114. <https://doi.org/10.36358/JCE.2018.18.4.91>
- [14] Hong, M., & Cho, J. (2021). A Conceptual Study on Computational Thinking for Non-majors. *Journal of Convergence for Information Technology*, 11(10), 151-158. <https://doi.org/10.22156/CS4SMB.2021.11.10.151>
- [15] Jeon, Y. (2018). *A Study on Utilization of Bebras Challenge Tasks as a Measurement Tool for Computational Thinking*. Proceedings of the Winter Academic Conference of the Korean Association of Computer Education, Chungbuk.
- [16] Park, H., & Jeon, Y. (2021). A Design of SW Liberal Arts Class to Cultivate Creative Problem-Solving Ability and Learning Motivation of Non-SW Majors:Based on the CT-CPS Instructional Model. *Proceedings of the Korean Computer Education Association Conference*, 25(2 (A)), 7-10.
- [17] Park, H. (2023). *The Development and Application of a MCPS(Motivation and Creative Problem Solving) Instructional Model for Computing Liberal Arts for Non-Majors*. [Doctoral dissertation], Andong National University
- [18] Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc
- [19] Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35
- [20] Wing, J. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717-3725
- [21] Kim, H., Seo, Y., Song, J., & Lee, J. (2020). The Direction of Future Elementary and Middle School SW and AI Education seen through Computing Curricula 2020. *Software Policy Research Institute. ISSUE REPORT*, 2020.12.10. IS-110
- [22] ACM & IEEE-CS. (2020). Computing Curricula Report. (2020). *Computing Curricula 2020: Paradigms for*

- Future Computing Curricula*. New York. <https://doi.org/10.1145/3467967>
- [23] Ministry of Science and ICT. (2023). *Announcement of Selection of Software-centered Universities for 2023*. Notice No. 2023-0163
- [24] Song, O. (2020). *The Effect of Software Education using micro:bit on Computational Thinking of Elementary School Students* [Doctoral dissertation], Gyeongsang National University
- [25] Kim, W. (2019). Exploring the Direction of Granular Basic-software Education Considering the Major of College Students. *JOURNAL OF The Korean Association of information Education*, 23(4), 329-341. <https://doi.org/10.14352/jkaie.2019.23.4.329>
- [26] Chung, H. (2021). Analysis of Algorithm Thinking based on Basic Software Education Effectiveness for non-major students in SW college. *Korean Journal of General Education*, 15(2), 175-187. <https://doi.org/10.46392/kjge.2021.15.2.175>
- [27] Kim, Y. (2020). *The Development and Application of Computational Thinking based Test Driven Problem Solving(CT-TDPS) Learning Model for Problem Solving Programming Education*. [Doctoral dissertation], Korea National University of Education
- [28] Kim, H., & Yun, B. (2020). A Study on Teaching-Learning Model Development for Problem Solving based on Computational Thinking. *Journal of Digital Contents Society* 21(5), 865-873. <https://doi.org/10.9728/dcs.2020.21.5.865>
- [29] You, J. (2022). The effect of SW project education using App Inventor on the computational thinking-oriented problem-solving skills and perceived SW learning task value: For freshmen in the humanities and social sciences. *The Journal of Korean Association of Computer Education*, 24(5), 1-16. <https://doi.org/10.32431/kace.2022.25.5.003>
- [30] Koh, E., & Lee, J. (2022). Design Guidelines for Python Programming Learning Support System for Non-Majors based on Computational Thinking. *The Journal of Educational Information*. 28(3), 733-762. <https://doi.org/10.15833/KAFEIAM.28.3.733>
- [31] Kim, J., et. al, (2015). *SW Education Teaching and Learning Model development Research*. KEDI & KERIS. CR2015-35
- [32] Korea Bebras Challenge. (2024, Apr 1). <https://www.bebas.kr/>
- [33] Bebras. (2024, Apr 1). <https://www.bebas.org/>
- [34] Jung, U., et. al. (2018). A Study on the Factors Influencing Computational Thinking Ability of Secondary School Students in Bebras Challenge 2017. *The Journal of Korean association of computer education*, 21(6), 21-33. <https://doi.org/10.32431/kace.2018.21.3.003>
- [35] Lee, J. (2020). Data science-based Python big data analysis, Hanbit Academy.
- [36] Jeon, Y. (2017). *The development and application of a CT-CPS(computational thinking-based creative problem solving) instructional model for the software education of new curriculum*. [Doctoral dissertation], Korea National University of Education
- [37] Seo, E. (2022). Design of SW Liberal Arts Education to Improve Learning Motivation for Computer Non-major - Focusing on the use of computing design think-up. *Journal of Digital Contents Society*, 23(11), 2171-2179. <https://doi.org/10.9728/dcs.2022.23.11.2171>
- [38] Kim, J., & Kim, D. (2024). Development and Application of AI Convergence Education Program Focus on Mathematics in Middle School. *The Journal of Korean Association of Computer Education*, 27(5), 35-46. <https://doi.org/10.32431/kace.2024.27.5.004>
- [39] Choi, H., & Kim, K. (2024). A Comparative Analysis of the Educational Effects of Block-type Programming Languages and Text-type Programming Languages. *The Journal of The Korean Association of Information Education*, 28(4), 387-397. <https://doi.org/10.14352/jkaie.2024.28.4.387>
- [40] Seo, J., & Shin, S. (2023). Exploring the Effectiveness of Major-Friendly SW Basic Education. *Journal of Digital Contents Society*, 24(3), 541-549. <https://doi.org/10.9728/dcs.2023.24.3.541>
- [41] Choi, S. (2019). *The Development of Multiple Project Based Coding Education Model focused on Computational thinking*, [Doctoral dissertation], Kyunghee University.
- [42] Kang, E. (2019). Structural Software Education Model for Non-majors-Focused on Python. *Journal of Digital Contents Society*, 20(12), 2423-2432. <https://doi.org/10.9728/dcs.2019.20.12.2423>



박희정

- 2001년 안동대학교 컴퓨터공학과(공학사)
- 2003년 안동대학교 컴퓨터교육과(교육학석사)
- 2023년 안동대학교 창의소프트웨어학과(공학박사)
- 2004년 ~ 현재 안동대학교 강사

✚ 관심분야 : 정보(SW-AI) 교육, 컴퓨팅 사고력, 정보 영재 교육, 비전공자 컴퓨팅 교육

✉ happy@anu.ac.kr



전용주

- 2005년 한국교원대학교 컴퓨터교육과(교육학학사)
- 2014년 한국교원대학교 컴퓨터교육과(교육학석사)
- 2017년 한국교원대학교 컴퓨터교육과(교육학박사)
- 2018년 ~ 현재 안동대학교 컴퓨터교육과 부교수

✚ 관심분야 : 정보(SW-AI) 교육, 컴퓨팅 사고력, 비버 챌린지, 정보 영재 교육, 인공지능융합교육

✉ yyongju@anu.ac.kr

부록

<표 1>. MCPS 수업모형의 단계

수업 단계	세부 수업 단계
단계 1. 동기 부여 및 준비 (Motivation)	1.1 학습동기 부여(전공과의 연관성, SW의 필요성) 1.2 학습자 특성 및 SW 교육의 어려움 파악(온라인 설문) 1.3 SW 교육의 선행요소 확인
단계 2. 기초지식 형성 (Building)	2.1 학습동기 부여(최신 IT 이슈로 관련성 찾기) 2.2 시범(프로그래밍 문법 설명) 2.3 지원단계(실생활 예제 활용) 2.4 도움중지 단계(실습 확장하기)
단계 3. 기초지식 응용 (Application)	3.1 학습동기 부여(SW기술 소개로 기대감 형성) 3.2 문제의 설정 3.3 문제 해결안 도출 3.4 해결 방법 공유
단계 4. 산출물 기획 및 제작 (Production)	4.1 아이디어 탐색 4.2 아이디어 구상 4.3 아이디어 설계 및 공유 4.4 아이디어 구현 4.4.1 아이디어 교정 4.4.2 산출물 공유 4.5 산출물 평가
단계 5. 발전 (Development)	5.1 학습동기 부여(도전 의식과 자신감 고취 확인) 5.2 디지털 테크놀로지 활용 5.3 다양한 매체를 통한 아이디어 표현

<표 8>. MCPS 평가 루브릭의 평가 요소

MCPS 수업 단계	지식	기능	태도	평가 요소
단계 1. 동기 부여 및 준비	○			1.1 컴퓨터의 기본원리와 개념
	○	○		1.2 전공 분야와 컴퓨팅의 관련성 인식
	○	○	○	1.3 실생활의 컴퓨팅 알고리즘 설계
단계 2. 기초지식 형성	○			2.1 프로그래밍 요소의 개념
	○	○		2.2 프로그래밍 요소를 이용한 코드 실행
	○	○	○	2.3 실생활 문제의 해결안을 위한 코드 실행
단계 3. 기초지식 응용	○			3.1 실생활(전공관련) 문제해결을 위한 코드 분석
	○	○		3.2 실생활(전공관련) 문제해결을 위한 코드 실행
	○	○	○	3.3 문제해결안 표현
단계 4. 산출물 기획 및 제작	○			4.1 실세계의 문제 탐색
	○	○		4.2 자료수집, 분석의 적절성
	○	○		4.3 문제를 해결 가능한 단위로 분해
	○	○		4.4 문제해결안 설계
	○	○		4.5 문제해결안 구현
	○	○		4.6 테스트와 디버깅을 통한 수정
	○	○	○	4.7 공유와 협력을 통한 피드백
	○	○	○	4.8 협력적 산출물의 결과 표현
단계 5. 발전	○	○	○	5.1 아이디어의 다양한 표현
	○	○	○	5.2 컴퓨팅 수업 경험 및 참여 소감 공유