

문제해결력 향상을 위한 비전공자 소프트웨어 기초교육 내용 분석 - 국내 SW중심대학 중심으로 -[☆]

Contents Analysis of Basic Software Education of Non-majors Students for Problem Solving Ability Improvement - Focus on SW-oriented University in Korea -

장은실¹ 김재현^{2*}
Eunsill Jang Jaehyou Kim

요 약

정부는 2015년부터 소프트웨어 중심대학 지원 사업을 통하여 미래 인재에 요구되는 소프트웨어 역량 강화에 힘쓰고 있다. 소프트웨어 중심대학으로 선정된 대학에서는 각기 다른 전공지식과 소프트웨어 소양을 겸비한 융합형 인재 양성을 위하여 대학 내 인문, 사회, 공학, 자연과학, 예체능 등 모든 계열에 소프트웨어 기초교육을 실시하고 있다. 본 논문에서는 20개 소프트웨어 중심대학에서 실시하고 있는 비전공자 대상의 소프트웨어 기초교육의 내용을 분석하였다. 분석 결과, 비전공자 학생들에게 실시하고 있는 소프트웨어 기초교육의 대부분은 미래사회에 필요한 컴퓨팅 사고력 중심의 문제해결력 향상과 컴퓨터과학에 기반을 둔 융합 능력 향상 목적으로 실시하고 있었다. 전공별 특성을 반영한 교육 내용과 프로그래밍의 난이도 조정을 위해 블록 기반 교육용 프로그래밍 언어와 텍스트 기반 고급 프로그래밍 언어를 활용하고 있다. 문제해결을 위한 교수 학습 방법으로는 문제 중심 학습(Problem based Learning), 프로젝트 중심 학습(Project Based Learning)과 토의·토론법을 많이 사용하고 있는 것으로 분석되었다. 향후 이 논문이 비전공자 소프트웨어 기초교육의 체계적 방향 설정에 도움이 되었으면 한다.

☞ 주제어 : 비전공자 소프트웨어 기초교육, 융합적사고력, 창의적사고력, 컴퓨팅사고력, 문제해결, 소프트웨어 기초교육, 소프트웨어 중심대학

ABSTRACT

Since 2015, the government has been striving to strengthen the software capabilities required for future talent through software-oriented university in Korea. In the university selected as a software-oriented university, basic software education is given to all departments such as humanities, social science, engineering, natural science, arts and the sports within the university in order to foster convergent human resources with different knowledge and software literacy. In this paper, we analyze the contents of basic software education for twenty universities selected as software-oriented universities. As a result of analysis, most of the basic software education which is carried out to the students of the non-majors students was aimed at improvement of problem solving ability centered on computational thinking for future society and improvement of convergence ability based on computer science. It uses block-based educational programming language and text-based advanced programming language to adjust the difficulty of programming contents and contents reflecting characteristics of each major. Problem-based learning, project-based learning, and discussion method were used as the teaching and learning methods for problem solving. In the future, this paper will help to establish the systematic direction for basic software education of non-majors students.

☞ keyword : Basic Software Education of Non-majors Students, integrated Thinking, Creative Thinking, Computational Thinking, Problem Solving, Basic Software Education, Software-oriented University in Korea

1. 서 론

1 Sungkyun Software Education Institute, Sungkyunkwan University., Seoul, 03063, Korea.

2 Dept. of Computer Education, Sungkyunkwan University., Seoul, 03063, Korea.

* Corresponding author (jaekim@skku.edu)

[Received April 17, 2019, Reviewed April 25, 2019 (June 9, 2019 R2), July 3, 2019]

☆ 본 논문은 2018년도 한국인터넷정보학회 추계학술대회 우수 논문 추천에 따라 의 내용을 확장 및 수정한 논문임.

산업혁명의 시작으로 수학과 과학의 학문적 가치가 높아졌고, 곧 도래될 4차 산업혁명 시대를 대비하기 위하여 소프트웨어 코딩 교육 열풍이 일게 되었다. 세계 각 국에서의 산업이 소프트웨어 중심으로 변화함에 따라 소프트웨어 교육의 중요성을 깨닫게 되었고, 국가 차원에서 소프트웨어 교육을 강화하기 시작했다. 핀란드에서는 초등

학교부터 필수 이수 과목으로 프로그래밍 교육을 지정하였고, 프랑스에서는 중학교 정규 과목으로 소프트웨어 교육을 지정하였으며, 영국에서는 초·중·고등학교 필수 이수 과목으로 컴퓨팅(computing) 교육을 지정하였다. 미국에서는 각 주마다 약간씩 다르지만 정규 교육과정으로 컴퓨터과학 과목을 채택하기 시작했으며, 이외의 다른 여러 나라에서도 소프트웨어 중심 세상에서 요구하는 인재 양성 및 역량 강화를 위하여 소프트웨어 교육은 필요한 교육이 되었다[1].

이러한 세계적 변화에 맞는 경쟁력을 갖추기 위하여 우리나라에서도 2015년 개정 교육과정을 통하여 소프트웨어 교육을 강화하기 시작하였다. 2019년도부터 초등학교 5~6학년 실과에서 17시간의 소프트웨어 교육이 의무화되고, 중학생에게는 2018년도부터 정보과목을 통하여 최소한 34시간 이상의 소프트웨어 교육이 필수로 도입되었으며, 고등학교에서는 심화선택인 정보과목을 일반선택으로 전환하였다[2].

교육의 변화는 초·중학교 외에도 과학기술정보통신부와 정보통신기술진흥센터가 공동으로 지원하는 소프트웨어 중심대학을 통하여 2015년도부터 대학교 교육에도 변화하기 시작하였다. 소프트웨어 중심대학으로 선정된 대학교에서는 산업현장에서 요구하는 창의적인 문제해결 역량을 갖춘 소프트웨어 전문 인력 양성을 위한 교육으로 교과과정을 전면 개편하였다[2][3].

소프트웨어 중심대학에서는 소프트웨어 전문 인력 양성을 위한 교육 외에도 비전공자 대상의 소프트웨어 융합 인재 양성을 위한 교육도 의무화했는데, 대학 내 소프트웨어 기초교육을 확대하여 모든 비전공자들의 전공별 특성을 반영한 소프트웨어 기초교육을 실시하였다[3].

이에, 본 논문에서는 2015년부터 2017년도까지 소프트웨어 중심대학으로 선정된 20개 대학의 비전공자 대상 소프트웨어 기초교육 내용을 분석한다. 분석을 통하여 타 전공지식을 가진 비전공자가 소프트웨어 소양을 겸비하여 창의·융합적인 문제해결을 할 수 있는 효율적인 소프트웨어 기초교육의 방향을 제시하고자 한다.

본 논문의 구성은 2장에서 문제해결을 위해 요구되는 사고력과 융합인재 양성을 위한 소프트웨어 소양 교육 및 기초교육에 대하여 소개하고, 3장에서는 비전공자 소프트웨어 기초교육 내용을 분석하기 위한 과정을 보이며, 4장에서는 문제해결을 위해 반영하고 있는 사고력들과 프로그래밍 언어 활용 및 교수·학습법들의 운영 현황을 분석한다. 그리고 5장에서는 결론 및 향후 연구과제에 대하여 제시한다.

2. 선행 연구 및 관련 연구

2.1 문제해결을 위한 사고력에 대한 선행 연구

미래인재위원회는 4차 산업혁명 시대에 요구하는 미래 인재상을 ‘창의성 기반의 문제해결력’, ‘기계와 공생을 통한 대안 도출 능력’, ‘기계와 협력하는 소통 능력’을 가진 인재로 정의했다[4].

OECD가 1997년 제시한 생애 핵심 역량에서는 ‘도구의 상호 작용적 활용’, ‘이질적인 그룹과의 사회적 상호 작용’, ‘자율적 행동’ 역량을 미래 사회에서 갖추어야 할 3대 핵심역량으로 제시했다[5].

미래 사회에서는 창의적으로 문제를 해결할 수 있는 인재상을 요구하고 있기 때문에 문제해결력 향상을 위하여 다양한 형태의 사고력 중심의 교육들이 진행되고 있다. 산업혁명 이후에는 수학과 과학을 통하여 논리적 사고를 향상시키기 위한 교육을 강화하였고, 곧 다가올 4차 산업혁명을 대비해서는 소프트웨어 교육을 통하여 창의·융합적 사고력 중심의 종합적인 문제해결력을 갖춘 인재 양성으로 교육계의 패러다임이 변화하고 있다[6].

문제해결을 위한 사고 능력으로는 컴퓨팅 사고력, 논리적 사고력, 창의적 사고력, 융합적 사고력, 디자인 사고력 등이 있다. 컴퓨팅 사고력은 문제를 정의하고 그에 대한 답을 기술하는 과정에서 문제 분해, 패턴 인식, 추상화, 알고리즘 등의 단계를 거친다. 컴퓨팅 사고력을 통하여 크고 복잡한 문제를 알고리즘으로 절차화하고 컴퓨터로 자동화하여 해결할 수 있는 모든 여러 학문 분야와 융합하는 중요성에 대하여 강조하고 있다[7]. 논리적 사고력은 주어진 정보나 주장 등에 대하여 정확한 판단을 할 수 있는 비판적 사고 능력과 대상들 간의 관계와 모순 등의 일련의 규칙들을 판단하고 평가하는 과정에서 추리 능력을 강조하고 있다[8]. 창의적 사고력은 오랜 경험과 축적된 지식을 바탕으로 기존 지식과 상식을 넘어서는 역발상적 사고를 발휘하여 문제를 해결하는 능력으로써 창의적 발견을 위한 열정과 내적 동기에 대한 부분을 강조하고 있다[9]. 융합적 사고력은 이질적인 것들을 모아 새로운 가치를 창출하여 문제를 해결하는 능력으로써 어느 한 분야에 치중하지 않는 유연하면서도 개방적인 사고를 강조하고 있다[10]. 디자인 사고력은 문제에 대한 실용적이고 창의적인 해결을 위해 문제에 대한 공감, 문제 정의, 아이디어 도출, 프로토타입 제작, 테스트 등의 5단계를 거친다. 디자인 사고력은 알려지지 않은 상황과 대안적 해결을 찾기 위하여 문제 상황을 직접 관찰하고 경

협하는 공감 과정을 거쳐 문제들을 정의하고 아이디어를 도출하는 과정을 강조하고 있다[11].

2.2 융합인재 양성을 위한 소프트웨어 소양 교육

곧 도래될 4차 산업혁명의 미래사회에서는 소프트웨어 교육이 해당 학문을 전공해야만 접할 수 있는 분야가 아니게 된다. 이미 소프트웨어는 다양한 분야와 융합되어 새로운 패러다임들로 쏟아져 나오고 있다. 따라서 인문, 사회, 공학, 자연과학, 예체능 등의 모든 분야에서 소프트웨어 소양을 겸비한 능력이 요구되고 있고, 이는 모든 분야에 걸쳐 소프트웨어 소양을 겸비한 융합형 인재를 양성해야 한다는 것이다. 초등학교 교육에서부터 시작되는 세계적인 코딩 교육 열풍은 미래사회에서 요구하는 창의적인 융합형 인재를 양성하기 위하여 이루어져야 하는 교육인 것이다[6].

2.3 국내대학 비전공자 소프트웨어 기초교육

소프트웨어 교육의 중요성이 알려지면서 전 세계는 소프트웨어 코딩 교육을 강화하기 시작하였다. 우리나라에서도 2015년 개정 교육과정을 통하여 초·중·고 교육에 소프트웨어 교육을 필수 또는 선택으로 편성하여 의무화 하였으며, 대학교에서는 소프트웨어 중심대학으로 선정된 대학들을 통하여 창의·융합적 사고력 향상을 위하여 비전공자에게도 소프트웨어 기초교육을 실시하기 시작하였다[2][3].

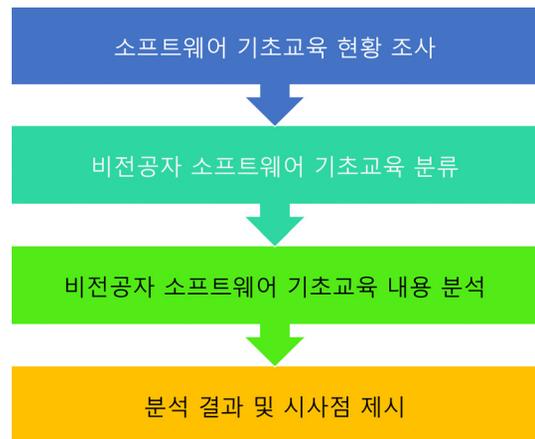
소프트웨어 중심대학으로 선정된 대학에서는 비전공자를 포함한 전교생에게 소프트웨어 기초교육을 실시해야 하는 의무가 있다. 그러나 비전공자 학생들에게는 동기부여 없이 소프트웨어 교육을 실시하게 되면 수업의 참여도와 교육의 효과가 떨어지게 된다[12][13]. 또한, 비전공자 대상의 소프트웨어 기초교육 내용은 컴퓨터과학 기초교육에 국한되지 않도록 각 전공분야에 직면한 문제해결을 컴퓨터과학 원리를 활용하여 문제에 접근하여 분석하고 해결하는 역량을 제고하고 있으며, 이를 실제로 체험하는 프로그래밍 실습 교육을 실시하도록 소프트웨어 중심대학에게 권고하고 있다[3][6].

즉, ICT 계열을 전공하지 않는 비전공자에게 진행하고 있는 소프트웨어 기초교육은 문제해결이 가능한 다양한 사고 능력 향상이 주요한 교육 목적이 되어야 한다. 따라서 ICT 계열 전공자들에게 교육하고 있는 방식과는 다르게 소프트웨어 교육을 진행해야 되는데, 긍정적으로 소프

트웨어 교육을 받을 수 있는 내적 동기부여를 밑바탕으로 전공별 특성을 반영한 교육 내용과 실습 프로그램이 구성되어야 한다. 소프트웨어 중심대학으로 선정된 각 대학에서는 특정 프로그래밍 언어(예: C, Python 등)의 문법 교육에 국한되지 않도록 사고력 중심의 창의적인 문제해결을 위한 소프트웨어 기초교육을 다양화하고 있다[14][15][16].

3. 연구 설계

본 연구는 2015년, 2016년, 2017년에 선정된 20개 소프트웨어 중심대학의 소프트웨어 기초교육 내용을 분석하고, 이를 통하여 비전공자 대상의 소프트웨어 기초교육에 대한 방향성을 제시하는 것을 목적으로 한다. 이를 위한 연구의 절차는 다음 그림 1과 같다.



(그림 1) 연구 절차
(Figure 1) Research procedure

20개 소프트웨어 중심대학의 비전공자 소프트웨어 기초교육 내용 분석을 위하여 필요한 현황 조사는 각 대학교 홈페이지에 제공된 강좌 정보를 활용하였다. 본 연구는 비전공자 소프트웨어 기초교육을 대상으로 하기 때문에, 다음과 같은 기준에 의해 분석 대상 강좌를 재 선별하였다.

첫째, 비전공자 대상의 소프트웨어 기초교육으로 확인할 수 있는 강좌를 대상으로 한다.

둘째, 비전공자 대상 이외의 소프트웨어 기초교육은 제외한다.

셋째, 비전공자 대상의 소프트웨어 기초교육이더라도

융합 또는 연계전공을 위한 소프트웨어 기초교육은 제외한다.

위의 세 기준에 따라 수집된 비전공자 대상의 소프트웨어 기초교육 강좌를 다시 재분류한 결과 71개 강좌가 선택되었고, 본 연구에서는 이 강좌들을 연구 대상으로 선정하여 소프트웨어 기초교육 내용을 분석하고, 분석 결과 및 시사점을 제시한다.

4. 연구 결과

집단 창조성이 뛰어난 인재 양성을 위해 ‘사고력 중심의 문제해결력’, ‘컴퓨터 프로그래밍 언어를 통한 기계와의 소통 능력’, ‘타 전공 분야와의 협력 능력’ 등을 20개 소프트웨어 중심대학의 비전공자 소프트웨어 기초교육 내용 분석을 통하여 확인하고자 한다.

첫째, 20개 소프트웨어 중심대학의 소프트웨어 기초교육 내용의 주요 구성을 분석하였다.

둘째, 문제해결을 위해 반영하고 있는 사고력 중심의 소프트웨어 기초교육 내용을 분석하였다.

셋째, 기계와의 소통 능력을 위한 컴퓨터 프로그래밍 언어들의 활용 현황을 분석하였다.

넷째, 타 전공 분야와의 협력 능력을 위한 교수·학습법 운영 현황을 분석하였다.

4.1 대학별 비전공자 소프트웨어 기초교육 현황

2015년, 2016년, 2017년에 선정된 20개 소프트웨어 중심대학의 비전공자 소프트웨어 기초교육 강좌의 강의 계획서 및 교육목표를 기반으로 정리한 교육 내용의 주요 구성은 다음 표 1과 같으며, 대학명은 정보보호 차원에서 알파벳으로 표기한다. 앞서 2.3 국내대학 비전공자 소프트웨어 기초교육에서 언급한 동기부여와 컴퓨터과학 원리를 활용한 문제해결의 교육 목적에 중점을 두고 교육 내용을 분석하였다.

4차 산업혁명 시대에서 요구하고 있는 미래 인재상의 중요한 역량은 사고력 중심의 문제해결 능력이기 때문에 20개 소프트웨어 중심대학의 대부분이 비전공자 소프트웨어 기초교육 목적으로 문제해결력 향상에 두고 있다는 것을 표 1에서 확인할 수 있었고, 비전공자의 수업 참여도와 교육의 효과를 높이기 위한 동기부여에 대한 교육은 20개 대학 중에서 12개 대학에서만 구성하고 있었다.

각 대학에서 문제해결력 향상을 위하여 주요하게 반영하고 있는 사고력은 컴퓨터사고력이 15건으로 가장 많았

고, 그 다음으로는 논리적 사고력과 창의적 사고력, 융합적 사고력, 디자인 사고력 순으로 나타났는데 표 1에서 확인할 수 있다. 4.2에서는 각 대학에서 반영하고 있는 사고력 중심의 소프트웨어 기초교육 내용을 분석한다.

4.2 사고력 중심의 소프트웨어 기초교육 내용 분석

4.2.1 컴퓨팅 사고력

컴퓨팅 사고력 기반으로 문제해결 능력을 교육하는 대학은 15개였다. 컴퓨팅 사고력 기반으로 문제해결을 교육하는 대부분의 15개 대학에서는 컴퓨팅 사고력 기반의 하위 요소를 제시하고 다양한 생활 속의 문제들을 요소에 적용 후 해결하는 과정을 절차화하여 표현하며, 컴퓨터 프로그래밍 언어를 이용하여 자동화하는 교육을 실시하고 있었다.

15개 대학 중에서 11개 대학에서는 자료구조와 알고리즘 기법들을 교육 내용에 추가적으로 구성하여 자동화하는 과정의 효율성을 높이기 위한 교육도 진행하였다.

그러나 컴퓨터과학 교육에 치우치지 않기 위해서는 컴퓨팅 사고력을 기반으로 하는 실생활의 문제해결에 대한 다양한 연구가 필요할 것으로 보인다.

4.2.2 논리적 사고력

논리적 사고력 기반으로 문제해결을 교육하는 대학은 4개였다. 논리적 사고력 기반으로 문제해결을 교육하는 4개 대학 중에서 2개 대학은 다양한 문제들을 컴퓨팅 사고력 단계에 적용하여 해결하는 과정에서 논리적 사고력 기반으로 절차화하여 표현하도록 교육하였으며, 나머지 2개 대학은 컴퓨터 프로그래밍 언어로 자동화 하는 단계에서 절차화 하는 과정을 논리적 사고력 기반으로 교육하였다. 특히, 4개 대학 중에서 1개 대학에서만 알고리즘을 교육하고 있었는데, 알고리즘 교육을 통하여 문제해결을 위한 알고리즘 설계 역량을 배양하도록 교육을 진행하였다.

논리적 사고력과 컴퓨팅 사고력 기반으로 문제해결을 교육하는 1개 대학에서는 전공에 따라 교육 내용을 다양화 했는데, 이공계열 학생들에게 자료구조와 알고리즘 기법들의 교육 내용을 추가적으로 구성하여 빅데이터, 인공지능 분야에 활용하는 교육도 진행하고 있었다.

4.2.3 창의적 사고력

창의적 사고력 기반으로 문제해결을 교육하는 대학은 4개였다. 창의적 사고력 기반으로 문제해결을 교육하는 4

(표 1) 비전공자 소프트웨어 기초교육의 주요 내용

(Table 1) Major Contents for Basic Software Education of Non-majors Students

대학	동기부여	교육 목적	교육 내용	사고력
A(17)	생활 속 문제해결 제시	문제해결력 향상	프로그래밍기초, 자료구조, 알고리즘	컴퓨팅 사고력 창의적 사고력
B(18)	소프트웨어와 변화하는 사회	문제해결력 향상	프로그래밍기초, 알고리즘	컴퓨팅 사고력
C(19)	활동 기반 생활 속 문제해결	창의적 문제해결력 향상	프로그래밍기초, 알고리즘	창의적 사고력 디자인 사고력
D(20)	소프트웨어와 변화하는 사회	창의적 감성 능력 함양	프로그래밍기초, 알고리즘	컴퓨팅 사고력 융합적 사고력
E(21)(22)	생활 속 ICT 기술	문제해결력 향상	프로그래밍기초, 알고리즘	컴퓨팅 사고력
F(23)(24)	—	ICT 융합 기초 능력	프로그래밍기초	논리적 사고력 융합적 사고력
G(25)	—	융합적사고 능력 향상	프로그래밍기초	융합적 사고력
H(26)	시대변화와 소프트웨어 기술	창의적 인재 양성	프로그래밍기초, 자료구조	컴퓨팅 사고력
I(27)	일상생활에서의 프로그래밍	논리적 문제해결력 향상	프로그래밍기초, 알고리즘	컴퓨팅 사고력 논리적 사고력
J(28)	—	융합적사고 능력 향상	프로그래밍기초	논리적 사고력 창의적 사고력
K(29)	—	문제해결력 향상	프로그래밍기초, 알고리즘	컴퓨팅 사고력
L(30)	—	창의·융합적 사고력 배양	프로그래밍기초	컴퓨팅 사고력 창의적 사고력
M(31)(32)	소프트웨어가 변화시키는 사회	소프트웨어 활용 능력	프로그래밍기초, 알고리즘	컴퓨팅 사고력
N(33)	—	SW활용 문제해결력 배양	프로그래밍기초, 자료구조, 알고리즘	
O(34)	디지털 사회와 미래 역량	미래사회 대처 능력	프로그래밍기초	컴퓨팅 사고력
P(35)	—	문제해결력 향상	프로그래밍기초	컴퓨팅 사고력
Q(36)	소프트웨어 세상과 디지털 리터러시	창의·융합적 문제해결력 향상	프로그래밍기초, 자료구조, 알고리즘	컴퓨팅 사고력
R(37)	—	문제해결력 향상	프로그래밍기초	컴퓨팅 사고력 논리적 사고력
S(38)	소프트웨어가 변화시키는 세상	문제해결력 향상	프로그래밍기초, 알고리즘	컴퓨팅 사고력
T(39)	소프트웨어와 변화하는 사회	문제해결력 향상	프로그래밍기초, 알고리즘	컴퓨팅 사고력

개 대학 중에서 2개 대학에서는 팀을 구성하여 프로젝트를 수행함으로써 문제해결에 있어 창의성을 발휘할 수 있도록 교육하였으며, 나머지 2개 대학은 소프트웨어적 사고를 배양하는데 목적을 둔 교육을 실시하고 있었다.

특히, 창의적 사고력 기반으로 문제해결을 교육하는 1개 대학에서는 소프트웨어 기초교육을 진행하기 전에 팀을 구성하여 디자인 사고력 방법론 기반으로 팀원과 소통하고 협력하여 창의적 발견을 위한 열정과 내적 동기를 끌어올릴 수 있도록 교육 내용을 구성하였고, 전공에 따라 여러 프로그래밍 언어를 활용함으로써 창의적 발상을 다양한 형태로 표현할 수 있도록 교육하였다.

이외의 대학에서는 컴퓨터 프로그래밍 언어 교육에 치우치지 않고 창의성을 발휘하여 문제를 해결하는 교육이 될 수 있도록 다양한 연구가 필요한 것으로 보인다.

4.2.4 융합적 사고력

융합적 사고력 기반으로 문제해결을 교육하는 대학은 3개였다. 융합적 사고력 기반으로 문제해결을 교육하는 3개 대학 중 1개 대학에서는 지정 학과별로 2가지 형태의 교육을 구성하였는데, 첫 번째 형태는 다양한 문제를 컴퓨팅 사고력 하위 요소에 적용하여 해결하는 과정에서 융합적 접근 방식을 유도하는 교육이었고, 두 번째 형태는 과학(Science), 기술(Technology), 공학(Engineering), 예술(Arts), 수학(Mathematics) 등의 분야에 중점을 두고 과학과 수학의 개념과 원리를 토대로 공학과 기술을 통하여 실생활의 문제를 해결하는 STEAM 교육을 실시하고 있었다.

나머지 2개 대학에서는 학생들 본인의 전공과 융합하여 문제를 해결하는 프로젝트를 수행하고 발표함으로써 융합 교육을 진행하였다.

4.2.5 디자인 사고력

디자인 사고력 기반으로 문제해결을 교육하는 대학은 1개였다. 앞서 4.3.3 창의적 사고력에서 소개한 1개 대학에서 유일하게 디자인 사고력 기반으로 창의적 문제해결을 교육하고 있었다. 해당 대학에서는 디자인 사고력에서 강조한 문제 공감, 문제 정의, 아이디어 도출 과정을 위해 학기 초부터 팀을 구성하고 한 학기 과정 중에서 상당 시간을 편성하여 팀 활동을 진행함으로써 소통과 협력을 통한 창의적 문제해결이 가능하도록 교육을 진행하고 있었다.

4.3 프로그래밍 언어 활용 현황

비전공자 소프트웨어 기초교육에 전공별로 프로그래밍 언어를 다양하게 활용하면 추상적인 개념들을 이해시키고, 컴퓨터를 이용하여 실생활 문제를 다양하게 해결하게 됨으로써 창의·융합적 사고력 및 문제해결 능력을 향상시킬 수 있게 된다[8].

획일적으로 프로그래밍 언어를 선택하여 비전공자의 전공 특성을 고려하지 않는 소프트웨어 기초교육을 진행한다면 교육의 효율성이 떨어질 수밖에 없다. 20개 소프트웨어 중심대학마다 거부감은 줄이고, 흥미도는 높여 교육의 효율성을 높이기 위하여 계열별 또는 단과대학별 또는 학과별로 비전공자의 전공별 특성을 반영하여 다양한 프로그래밍 언어를 활용한 소프트웨어 기초교육을 실시하고 있었다.

표 2의 5개(C, E, I, N, S) 대학은 계열별 또는 단과대학별 또는 학과별로 소프트웨어 기초교육 강좌를 가장 많이 다양하게 개설한 대학으로써 각 비전공자 그룹에게 지정된 강좌를 수강하도록 했는데, 강좌에 대한 운영 현황을 정리하면 다음 표 2와 같다.

(표 2) 비전공자 전공별 소프트웨어 기초교육 강좌 현황
(Table 2) Basic Software Education Lecture according to Major of Non-majors Students

대학	비전공자그룹	개설 강좌수	수업 학기	주당 시수
C	계열별	5개 강좌	1, 2학기	오프 2시간
E	단과대학별	25개 강좌	1학기	오프 2시간
I	계열별	6개 강좌	1, 2학기	온 1.5, 오프 1.5시간
N	계열별	10개 강좌	1학기	오프 3시간
S	학과별	26개 강좌	1, 2학기	오프 3시간

비전공자 소프트웨어 기초교육에서 활용하고 있는 프로그래밍 언어들은 블록 기반 교육용 프로그래밍 언어, 다이어그램 언어, 텍스트 기반 고급 프로그래밍 언어 3종류로 구분할 수 있었다.

표 3에서의 3개(C, E, S) 대학은 단과대학과 학과별로 프로그래밍 언어를 지정하여 소프트웨어 기초교육을 실시하고 있었는데, 기본적인 3종류 언어 외에도 통계처리 및 데이터 시각화 소프트웨어를 활용하는 등으로 다양한 프로그래밍 언어를 활용하였다.

표 3에서의 4개(I, M, N, R) 대학은 2종류 이상의 언어들 중에서 계열별로 지정하여 프로그래밍 언어를 활용한 소프트웨어 기초교육을 실시하였다.

이외에 첫 번째 유형의 대학에서는 블록 기반 교육용

(표 3) 문제해결을 위한 프로그래밍 언어 활용
(Table 3) Usage of Programming Language for Problem Solving

대학	사용 언어 분류			
	블록 기반 교육용 프로그래밍 언어	다이어그램 언어	텍스트 기반 고급 프로그래밍 언어	기타
A			파이선, C	
B	스크래치		파이선, C	
C	스크래치, 블록리, 앱인벤터		파이선,	SAP 피오리, 매트랩
D	스크래치		파이선	
E	스크래치, 앱인벤터	랩터	파이선, C	매트랩, SPSS, R, GAS
F	스크래치, 앱인벤터		파이선, C, 프로세싱	
G	엔트리		파이선	
H	스크래치		파이선	
I	스크래치, 앱인벤터	랩터	파이선, C	R
J	엔트리			
K	엘리스		파이선	
L	스크래치, 스냅!		파이선, 자바, C, C++	
M	스크래치			HTML, SQL,
N			파이선, C, C++, 프로세싱	HTML, CSS, 자바스크립트, R
O	엔트리			
P			파이선, C	
Q	엔트리		파이선	
R			파이선, C	
S	스크래치		파이선, C	매트랩, 미니탭, SPSS, SAS, R, dbSTAT
T	스크래치		파이선, C	매트랩, R

프로그래밍 언어와 텍스트 기반 고급 프로그래밍 언어들을 활용한 개설 강좌 중에서 학생들이 선택하여 소프트웨어 기초교육을 받을 수 있도록 했다. 두 번째 유형의 대학에서는 프로그래밍 언어를 구분하지는 않지만 수업내용의 구성과 난이도 조정을 통하여 전공별 특성을 반영하려고 하였다. 이를 정리한 내용은 다음 표 3과 같다.

4.4 교수·학습법 운영 현황

문제해결을 위하여 비전공자에게 사고력 중심의 소프트웨어 기초교육을 실시한다는 것은 매우 어려운 문제인 것이 수업을 진행하면서 내용이 심화될수록 학습자의 집중도와 흥미도는 점점 낮아지는 경향이 드러나기 때문이다. 이에, 20개 소프트웨어 중심대학에서는 기존의 교수·학습법 외에도 효율적인 다양한 교수·학습법을 적극 활용하여 비전공자 대상의 문제해결을 위한 소프트웨어 기초교육을 실시하고 있었다.

20개 소프트웨어 중심대학 모두 시범 실습법으로 소프트웨어 기초교육을 실시하고 있었고, D2 대학을 제외한 19개 대학에서는 문제 중심 학습을 적용하여 전공별로 해결할 수 있는 다양한 문제해결을 위한 소프트웨어 기

초교육을 실시하고 있었으며, 11개(C, E, F, G, J, K, M, N, Q, S, T) 대학에서는 프로젝트 기반 학습을 적용하여 학습자 중심으로 문제해결을 위한 소프트웨어 기초교육을 실시하고 있었는데, 그 중에서 3개(E, K, Q) 대학에서는 멘토(선배 또는 조교)를 참여시켜 프로젝트를 심화 및 완성하도록 운영하고 있었다. 이외의 대학에서는 플립러닝·블렌디드러닝 교수·학습법으로 소프트웨어 기초교육을 실시하고 있었다. 이를 정리하면 다음 표 4와 같다.

5. 결론 및 향후 연구과제

2105년, 2016년, 2017년에 선정된 20개 소프트웨어 중심대학 대상으로 사고력 중심의 문제해결력 향상을 위하여 실시하고 있는 비전공자 소프트웨어 기초교육을 다양한 형태로 분석하였다.

비전공자 학생들에게 진행되고 있는 소프트웨어 기초교육은 미래사회에 필요한 인재 양성, 사고력 중심의 문제해결력 향상, 컴퓨터과학과의 연계학습을 위한 목적으로 실시하고 있었다.

20개 대학에서는 보다 나은 문제해결력 향상을 위하여 다양한 사고력을 적용하여 소프트웨어 기초교육을 실시하고 있었는데, 소프트웨어의 가치와 중요성을 이해하고 컴퓨터과학의 원리를 활용하여 문제를 분석하고 해결하는 컴퓨팅 사고력 중심의 교육이 가장 많았다.

인문, 사회, 공학, 자연과학, 예체능 등의 전공별 특성을 반영하기 위하여 20개 대학에서는 다양한 프로그래밍 언어를 활용하여 소프트웨어 기초교육을 실시하고 있었는데, 블록 기반 교육용 프로그래밍 언어와 텍스트 기반 고급 프로그래밍 언어가 두드러지게 많다는 것을 확인할 수 있었다.

미래 사회에서 요구하는 인재상은 소통과 협업이 중요한 요소인데, 이러한 요소를 반영하기 위하여 다양한 교수·학습법을 적용한 소프트웨어 기초교육을 확인할 수 있었고, 실습으로 이루어지는 교육이다 보니 20개 대학 모두 시범실습법 형태로 교육이 이루어졌고, 문제해결을 위한 문제 중심 학습, 프로젝트 기반 학습과 함께 토의·토론법이 주로 많이 적용되고 있었다.

본 논문에서 살펴본 20개 소프트웨어 중심대학에서는 비전공자 학생들에게 소프트웨어 기초교육에 대한 거부감을 최소화하고, 효율적인 교육이 되도록 다양한 형태로 강좌들을 운영하였으며, 2015년에 선정된 소프트웨어 중심대학보다는 이후에 선정된 대학들에서의 소프트웨어 기초교육이 좀 더 나아진 형태로 운영되어지면서 점점 안정화되어 가고 있다는 것을 확인하였다. 이는 학생들

(표 4) 대학별 문제해결을 위한 교수·학습법 운영 현황
(Table 4) Operational of Teaching Learning Methods for Problem Solving

대학	시범 실습법	문제 중심 학습	토의·토론법	프로젝트 기반 학습	플립러닝·블렌디드러닝
A	○	○			
B	○	○			○
C	○	○	○	○	
D	○	○			
E	○	○	○	○	
F	○	○	○	○	
G	○	○	○	○	
H	○				
I	○	○			
J	○	○	○	○	
K	○	○	○	○	
L	○	○	○		
M	○	○	○	○	
N	○	○	○	○	
O	○	○			
P	○	○			
Q	○	○	○	○	
R	○	○			○
S	○	○	○	○	
T	○	○	○	○	

스스로도 미래사회에서 요구하는 인재상에 대한 사회적 분위기를 통하여 내적 동기부여가 조금씩 형성되기 때문이라고 할 수 있겠다.

공개되어 있는 교육내용들 중심으로 분석하다 보니, 공개된 정보가 부족한 대학교들과 소프트웨어 중심대학 초기에 공개된 내용에서 개선된 교육내용들을 공개 반영하지 않은 대학교들은 실제 운영되고 있는 교육내용을 반영했다고 할 수 없다는 점에서 연구의 한계가 있었다.

향후 연구에서는 분석한 20개 소프트웨어 중심대학 외에 2018년~2019년에 선정된 15개 대학을 추가하여 총 35개 소프트웨어 중심대학의 비전공자 대상 소프트웨어 기초교육을 더욱더 심도 있게 분석하여 효율적인 비전공자 소프트웨어 기초교육을 체계화하고, 미래 사회에서 요구하는 소프트웨어 소양을 겸비한 융합인재 양성 교육에 도움이 되고자 한다.

참고문헌(References)

- [1] H. C. Kim, Examples of Foreign Software Education and Implications, Webzine of Korean Educational Development Institute (KEDI), 2018.
http://edzine.kedi.re.kr/2015/summer/article/world_02.jsp
- [2] 2015 revised curriculum summary, National Curriculum Information Center, 2018.
<http://ncic.go.kr/mobile.dwn.ogf.inventoryList.do>
- [3] Software-oriented University, Institute of Information & Communications Technology Planning & Evaluation (IITP), 2018.
<https://www.iitp.kr/kr/1/business/businessMap/view.it?seq=309>
- [4] H. Cho, Future Talent in the Age of the Fourth Industrial Revolution, UNIVERSITY NEWS NETWORK, 2018.
<http://news.unn.net/news/articleView.html?idxno=188303>
- [5] The Definition and Selection of Key Competencies: Executive summary, pp. 10-15, Organization for Economic Cooperation and Development (OECD), 2018.
<https://www.oecd.org/pisa/35070367.pdf>
- [6] SW-oriented University Training Program to Cultivate Creative SW talents for the Success of K-ICT Strategy, pp. 9-10, Ministry of Science and ICT (MSIT), 2018.
http://www.msip.go.kr/cms/www/open/go30/publicInfo/info/info_1/info_11/_icsFiles/afildfile/2015/08/11/2015년도%20SW중심대학추진계획.hwp
- [7] Aman Yadav, et al., "Introducing Computational Thinking in Education Courses," Proceedings of the 42nd ACM technical symposium on Computer science education, pp. 465-470, 2011.
<https://doi.org/10.1145/1953163.1953297>
- [8] What is logical thinking?, sheffield, 2019.
<https://sheffield.ru/ko/thinking/logical-thinking-what-is-logical-thinking/>
- [9] J. Y. Chang, Understanding the basics of creativity, ArtE Academy of KACES, pp. 9-16, 2019.
http://www.cultureline.kr/webgear/board_pds/9133/창의성의%20이해%20창의적%20사고력.pdf
- [10] The ability to understand fusion more than the amount of knowledge is essential, Blog of the Korean Ministry of Education, 2019.
<https://blog.naver.com/moeblog/221173835448>
- [11] Design Thinking, Wikipedia, 2019.
https://en.wikipedia.org/wiki/Design_thinking
- [12] 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.
<https://doi.org/10.14400/JDC.2017.15.7.123>
- [13] J. E. Nah, "Software Education Needs Analysis in Liberal Arts", Korean Journal of General Education, Vol. 11, No. 3, pp. 63-89, 2017.
<http://www.riss.kr/link?id=A103253836>
- [14] K. S. Oh, E. K. Suh, H. J. Chung, "A study on development of educational contents about combining computational thinking with design thinking," Journal of Digital Convergence, Vol. 16, No. 5, pp. 65-73, 2018.
<https://doi.org/10.14400/JDC.2018.16.5.065>
- [15] 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.
<http://dx.doi.org/10.9728/dcs.2017.18.7.1313>
- [16] M. J. Kim, H. C. Kim, "Effectiveness analysis based on computational thinking of a computing course for non-computer majors," The Journal of Korean Association of Computer Education, Vol. 21, No. 1, pp. 11-21, 2018.
<http://www.riss.kr/link?id=A105095555>

- [17] Software-oriented University Group of Kyunghee University, 2018. http://swedu.khu.ac.kr/html_2018/
- [18] Software-oriented University Group of Kwangwoon University, 2018. <http://npsw.kw.ac.kr/>
- [19] Software-oriented University Group of Dankook University, 2018. <http://swcu.dankook.ac.kr/>
- [20] Software Convergence Institute of Chosun University, 2018. <http://sw.chosun.ac.kr>
- [21] Da Vinci Software Education Institute, 2018. <https://sw.cau.ac.kr/index001>
- [22] Y. M. Kim, M. J. Lee, "A Comparative Study of Educational Programming Languages for Non-majors Students: from the Viewpoint of Programming Language Design Principles", The Journal of Korean Association of Computer Education, Vol. 22, No. 1, pp. 47-61, 2019. <https://doi.org/10.32431/kace.2019.22.1.005>
- [23] Institute of Software Education in Handong Global University, 2018. http://www.handong.edu/major/college_ITC/ISE/intro/
- [24] H. J. Kim, K. M. Kim, "Effect Analysis of the SW Curriculum Reorganization for Non-CS Students on the Class Satisfaction", Proceedings of the 2018 Korean Association of Computer Education Summer Conference, Vol. 22, No. 2, pp. 45-48, 2018. <http://www.riss.kr/link?id=A105507039>
- [25] Software-oriented University Group of Kookmin University, 2018. <https://www.facebook.com/kmucsssoftware>
- [26] Convergence Software Institute Dongguk, 2018. <https://csid.dongguk.edu/>
- [27] Pusan National University Software Education Center, 2018. <http://his.pusan.ac.kr/swedu/index.do>
- [28] Software-oriented University of Seoul Women's University, 2018. <http://software.swu.ac.kr/>
- [29] KAIST Software Education Center, 2018. https://academy.kaist.ac.kr/pages/sub/sub0102_9
- [30] Software-oriented University Group of Hanyang University, 2018. <http://computing.hanyang.ac.kr/about/agency8.php>
- [31] Software-oriented University of Gachon University, 2018. <http://xsw.gachon.ac.kr/cms/>
- [32] S. H. Park, "Study of SW Education in University to enhance Computational Thinking", Journal of Digital Convergence, Vol. 14, No. 4, pp. 1-10, 2016. <http://dx.doi.org/10.14400/JDC.2016.14.4.1>
- [33] Software Education Center of Kyungpook National University, 2018. <http://swedu.knu.ac.kr/>
- [34] Software-oriented University of Korea University, 2018. <https://helloworld.korea.ac.kr/>
- [35] Sogang University Software Education Center, 2018. <http://swedu.sogang.ac.kr/index.do>
- [36] Sungkyun Software Education Institute, 2018. <http://ssen.skku.edu/index.do>
- [37] Software-oriented University of Sejong University, 2018. <http://sw.sejong.ac.kr/index.do>
- [38] Ajou University Basic Software Education, 2018. <https://sites.google.com/a/ajou.ac.kr/sw-workshop/home>
- [39] Software-oriented University of Chungnam University, 2018. <http://wise.cnu.ac.kr/>

● 저 자 소 개 ●



장은실(Eunsill Jang)

2001년 동국대학교 교과교육학과 컴퓨터공학전공 (교육학석사)
2007년 동국대학교 일반대학원 컴퓨터공학과 (공학박사)
2008년~2011년 동국대학교 산업기술연구원 전임연구원
2016년~2018년 (주)명리 개발지원팀 이사
2018년~현재 성균관대학교 소프트웨어대학 초빙교수
2018년~현재 성균관대학교 성균SW교육원
관심분야 : 소프트웨어 교육, SW 융합교육, SW영재 융합교육, SW영재 담당교원 역량강화, etc.
E-mail : janges@skku.edu



김재현(Jaehyou Kim)

1988년 성균관대학교 수학과 졸업(학사)
1992년 웨스턴일리노이 주립대학교 대학원 전산학과 석사
2000년 일리노이공과대학교 대학원 전산학과 박사
2014년~현재 한국컴퓨터교육학회 부회장
2010년~현재 한국인터넷정보학회 부회장
2002년~현재 성균관대학교 컴퓨터교육과 교수
2016년~현재 성균관대학교 성균SW교육원 원장
관심분야 : 객체지향 소프트웨어공학, 컴퓨터교육, Computer Based LET etc.
E-mail : jaekim@skku.edu