

# IT 융합교육을 위한 비전공자 코딩교육의 발전방안

피수영

대구가톨릭대학교 교양교육원

## A Study on Coding Education of Non-Computer Majors for IT Convergence Education

Su-Young Pi

Dept. of Institute of Liberal Education, Catholic University of Daegu

요 약 코딩교육은 융합형 교육의 효과적인 도구로 문제해결 방법을 기획하고 설계하는 과정 등을 통해 문제해결 능력, 논리적 사고능력, 창의적 사고능력 등을 키울 수 있다. 코딩교육을 위한 연구는 주로 초.중등 학생들을 대상으로 연구가 활발히 진행되고 있으며 대학생들을 대상으로 한 연구는 부족한 실정이다. 지금 현시점에서는 코딩교육을 전혀 받지 못한 대학생들의 교육이 시급한 실정이다. 코딩교육의 필요성을 인지하고 대학에서도 교양필수로 교과목을 지정하여 교육하려는 시도를 하고 있지만 어떻게 교육을 할 것인지에 대해 고심하고 있는 실정이다. 보편적 교육으로서의 코딩교육에 대한 연구가 필요한 시점이다. 따라서 본 연구에서는 비전공자들을 대상으로 코딩교육을 효율적으로 운영하기 위해 플립러닝을 적용한 앱인벤터를 활용한 교육과정을 제시하였다. 교육의 효과를 측정하기 위해 설문조사와 개인 인터뷰를 실시하였다. 코딩교육을 통해 비전공자 학습자들의 전공영역과 융합하여 다양한 문제들을 컴퓨팅 사고력 기반으로 해결해 나갈 수 있는 문제해결 능력을 향상시킬 것으로 기대한다.

주제어 : 융합교육, 코딩교육, 컴퓨팅 사고력, 플립러닝, 앱인벤터

**Abstract** Coding education is an effective convergence type educational tool. While solving problems and designing programs, students can enhance problem solving ability, logical reasoning ability and creative thinking. Researches on coding education are done primarily for elementary school and middle school students. However, researches on college students are lacking. Today, educating college students about coding is in dire need. Although there are efforts to promote the importance of coding education and make it requirements. People find it difficult to find ways to provide training. There is a need for researches on coding as universal education. Therefore, this research proposed educational training using app inventor based on flipped running in order to effectively promote coding education. This study conducted the survey and the personal interview to measure the effectiveness of coding education. It is hoped that, through coding education, students who do not major in coding could combined their knowledge of their major with coding to improve their problem solving ability to solve various problems based on computing knowledge and approach.

**Key Words** : Convergence Education , Coding Education, Computational Thinking, Flipped Learning, App Inventor

\* 본 논문은 2015년 대구가톨릭대학교 교내연구비를 지원받아 수행하였음(과제번호 20151063)

Received 5 August 2016, Revised 5 September 2016

Accepted 20 October 2016, Published 28 October 2016

Corresponding Author: Su-Young Pi

(Catholic University of Daegu)

Email: sypi@cu.ac.kr

ISSN: 1738-1916

© The Society of Digital Policy & Management. All rights reserved. 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.

## 1. 서론

현대사회는 소프트웨어를 중심으로 한 융합기술사회로 변화해가고 있으며 다양한 분야의 전문지식을 갖춘 융복합 인재에 대한 수요가 날로 증가하고 있다. 최근 우리나라의 글로벌 경쟁력 중에서 가장 앞선 분야는 IT이다. 전 세계 IT산업 중 소프트웨어 산업은 지식집약 산업으로 부가가치가 높고 취업유발 효과도 매우 크므로 IT 융합교육에 대한 연구가 활발히 진행되고 있다[1]. IT 융합 교육의 핵심은 소프트웨어교육을 통해서 이루어지고 소프트웨어교육의 목적은 컴퓨팅 사고력(Computational Thinking)을 향상하는데 초점이 맞추어지고 있다[2,3,4]. 소프트웨어교육의 필요성은 변화되는 요구역량, 사고력 향상, 직업세계의 재편으로 분류할 수 있다. 새로운 디지털시대가 도래하고 학문과 산업에서 융복합적 성격을 띠는 분야가 많아지면서 컴퓨팅 사고력의 중요성이 강조되고 있다[5]. 컴퓨팅 사고력을 개발함으로써 비전공자들은 자신의 전공분야에서 IT를 통한 혁신 아이디어를 찾을 수 있는 능력과 혁신적인 아이디어를 구현할 수 있는 능력, IT 전문가와 소통할 수 있는 능력 등을 향상시킬 수 있다. 이러한 컴퓨팅 사고력은 컴퓨터 프로그래밍 언어 교육을 통해서 개발하고 훈련할 수 있다.

CT의 중요성이 강조되면서 이에 대한 연구 및 강화하기 위한 교육방법에 대한 연구도 활발히 진행되고 있다 [6,7,8]. 현재 국민대는 국내 최초로 전체 학부생을 대상으로 컴퓨터개론, 프로그래밍언어 등 1년간 6학점 이상 SW과목을 수강하도록 하였다. 연세대에서는 컴퓨팅사고력 과목을 2016학년도 신입생 전체에게 필수교과목으로 실시하고 있다[5]. 국내에서 이루어지고 있는 CT 교육 중 하나인 프로그래밍언어 교육은 문제를 분석하고 유형화하는 분석 작업과 알고리즘을 작성하는 과정을 통해 논리적 사고, 문제해결 역량, 창의력을 키울 수 있다. 안성진(2014)[9]의 ‘초중등 컴퓨터프로그래밍 교육과 논리적 사고력 향상과의 상관관계 연구’에 따르면 컴퓨터 프로그래밍교육을 통해 문제의 특성 및 해결책을 발견하고 설계하는 문제해결능력이 20.4% 향상 되었으며 문제요소들 간의 논리적 관계를 파악하고 해결책을 찾는 논리적 사고능력이 37.5% 향상이 되었다고 소프트웨어교육의 효과를 제시하였다. 이처럼 소프트웨어교육의 필요성이 대두되면서 초, 중등학교를 중심으로 교육 분야에

서도 활발히 움직이고 있는 상황이다. 하지만 현재 대부분의 대학에서 비전공자를 위한 컴퓨터교육이 소프트웨어 활용교육으로 파워포인트, 프레지, 엑셀, 포토샵, 멀티미디어 활용 등의 교육이 대부분이다[6,10]. 소프트웨어 중심대학으로 선정된 대학을 제외한 대학에서는 필요성을 인식하면서도 교육방향의 틀을 세워 실시하는 학교가 거의 없는 실정이다. 소프트웨어교육의 보편화를 위해 교양IT 교육과정을 프로그래밍 중심의 교과과정으로 개편할 필요가 있으며[5] 이공계열 학생들뿐만 아니라 인문사회, 예체능 계열의 모든 학생들까지 소프트웨어교육을 실시해야 한다. 대학에서 지금까지의 코딩교육은 전문가 양성을 목적으로 컴퓨터전공 관련 학과에서만 교과과정으로 진행하고 있어 보편적 교육으로서의 코딩교육에 대한 연구가 필요한 시점이다. 단순한 코딩교육이 아니라 일상생활 속에서 나타나는 문제들을 컴퓨팅 사고력 기반의 코딩교육을 통해 해결해 나가는 교육과정이 필요하다.

비전공자들의 소프트웨어교육을 위한 수업모형 개발이 절실히 요구되는 시점에 본 연구는 비전공자들을 위한 코딩교육 학습모형을 제시하고, 개발한 학습모형이 비전공자들의 전공영역과 융합하여 문제를 해결할 수 있는 능력을 향상시킬 수 있는지의 효과를 검증하는데 연구의 궁극적 목적이 있다. 비전공자들의 보편적 교육으로서의 코딩교육을 위해 블록형언어의 하나인 앱인벤터를 활용한 교육과정을 제시하였다. 수업진행만으로 학습자들의 코딩교육의 효과를 얻어내기가 어려우므로 수업과 연계된 비교과프로그램인 앱개발 경진대회를 실시하였다. 학습자들이 자기 주도적으로 지식을 습득하고 습득한 지식에 대한 논의와 평가를 위한 학습자중심의 수업환경으로 진행하기 위해 플립러닝(Flipped Learning)[11]을 3주 동안 실시하였으며 이를 비교과프로그램과 연계시켜 진행하였다. 수업용 의사소통 도구인 핑퐁앱(PingPong APP)을 플립러닝의 평가와 수업에 대한 이해도, 프로젝트심사 등에 활용하였으며 교육의 효과를 측정하기 위해 설문조사와 개인 인터뷰를 실시하였다. 제시한 교육과정을 통해 비전공 학습자들의 전공영역과 융합하여 단순한 문제에서부터 현실적으로 다루기 어려운 복잡한 문제에 이르기까지 해결하고자 하는 다양한 문제들을 컴퓨팅 사고력기반으로 해결해나갈 수 있는 능력을 향상시킬 것으로 기대한다.

## 2. 소프트웨어 교육

### 2.1 코딩교육의 필요성

CT의 중요성이 강조되면서 국내에서 이루어지고 있는 CT 교육은 첫째, 교육용 프로그래밍 언어를 사용한 교육과 둘째, 컴퓨터 없이 컴퓨터과학의 원리를 가르치는 언플러그드 교육과 셋째, 센서를 이용하여 물리세계와 프로그램을 연결하는 피지컬컴퓨팅 교육, 넷째, 다른 학문에서 CT를 활용하는 융합컴퓨팅으로 구분할 수 있다[12]. 본 연구에서는 교육용 프로그래밍언어를 사용한 비전공자들의 코딩교육에 대한 수업모형을 개발하고자 한다. 교육용 프로그래밍언어는 우선적으로 교육적인 목적을 갖고 제작된 언어이기 때문에 비전공자들이 보다 쉽게 접근하여 문제해결 방법을 알려줄 수 있는 언어이다. 즉 교육용 프로그래밍언어는 컴퓨터 프로그래밍이나 컴퓨터공학에 관심을 가지고 있지만 출발점을 찾기 힘든 학습자나 혹은 프로그래밍 학습에 어려움을 겪고 있는 전공자에게 적합한 도구라고 할 수 있다. 교육용 프로그래밍언어를 통해 문법적 지식, 컴퓨터시스템 지식을 최소화하여 로직개발, 관찰력 훈련 등을 하게 된다면 컴퓨터문제와 상관없이 수준 높은 문제를 해결할 수 있는 능력있는 학습자가 될 수 있다[13].

프로그래밍을 하는 과정에서 자신의 인지과정 즉, 문제이해와 적절한 문제해결 전략의 사용여부, 자신의 이해와 산물에 대한 모니터링 등의 작업을 하는 가운데 자신의 사고과정을 인지하게 된다. 프로그래밍 교육환경은 학습자의 내적동기를 신장시키고 지적 호기심과 학습에 대한 열정 등을 향상시킬 수 있다[12]. 코딩교육은 소프트웨어를 개발하기 위해서만 필요한 것이 아니다. 코딩을 작성하기 위해서는 문제를 분석하고 유형화하는 분석작업과 이를 창의적 방법으로 해결하는 알고리즘을 작성하는 과정을 통해 논리적 사고, 문제해결 역량, 창의력을 키울 수 있다. 대학에서 프로그래밍언어 자체에 대한 학습을 지양하고 컴퓨팅 사고력기반의 문제해결력을 기르는 것이 필요하다. 프로그래밍언어는 핵심도구로서 활용이 이루어져야 한다[4,5,7]. 따라서 교양교과목으로 비전공자들 대상으로 컴퓨팅 사고능력 배양을 위한 프로그래밍 중심 교과과정을 개설하여 다양한 융합교육 모델의 기반을 제공할 필요가 있다. 이는 단순히 프로그래밍능력 이상의 비판적 사고와 문제해결능력을 동시에 갖춘

현재 대학사회에서 양성하고자하는 인재상과 공통적인 부분이다[14,15].

### 2.2 블록형 언어 앱인벤터

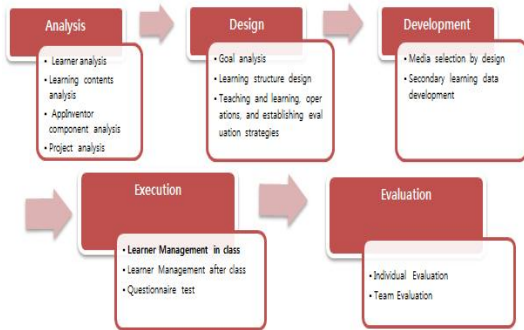
블록형 언어는 문제를 해결하기 위해 블록의 명령어를 결합하여 쉽고, 자유롭게 원하는 프로그램을 만들어 해결할 수 있다. 스크래치(Scratch), 앱인벤터(App Inventor), 엔트리(Entry) 등이 여기에 속한다. 앱인벤터2는 전문적인 프로그래밍 지식이 없는 비전공자나 컴퓨터 초보자도 웹브라우저를 통해 안드로이드폰 앱을 손쉽게 만들 수 있도록 도와주는 일종의 앱 개발도구이다. 현재는 미국 MIT대학에서 교육용 프로젝트로 운영하고 있는 오픈소스 프로그램이다[16]. 스마트폰 앱을 개발하기 위해서는 Java나 C언어 같은 프로그래밍언어를 배우는데 많은 노력을 해야 한다. 하지만 앱인벤터는 프로그래밍언어를 배우는 시간을 줄이고, 앱을 개발하는 과정 중에 생기는 문제들을 논리적으로 해결하는 과정을 겪으면서 창의력과 문제해결력을 키울 수 있도록 한다. 기존 프로그래밍언어와 달리, 직관적으로 눈으로 보면서 조합하여 프로그램을 완성해가는 방식이므로 쉽게 프로그래밍의 개념을 익히고 활용할 수 있는 환경을 제공한다.

Jenkins(2002)[17]는 프로그래밍의 어려움을 7가지로 분류하였다. 첫째, 다양한 기술로 Syntex 기본, 점차 시맨틱, 구조, 스타일로 옮겨감. 둘째, 다양한 과정으로 알고리즘을 세분화하는 과정, 그것을 코드로 옮기는 과정이다. 셋째는 언어로 프로그래밍언어 문제를 어렵게 여겼다. 넷째는 교육적인 새로움으로 새로운 과목에 대한 부담감을 들었다. 다섯째는 관심영역으로 학생들은 지루해 하므로 최선의 프로그래밍은 재미있고, 창의적인 활동으로 구성해야 한다고 했다. 여섯째는 평판과 이미지 영역으로 어렵다는 이미지이고, 마지막으로 대학에서 배우기 때문에 시간 스케줄의 문제로 7가지 영역의 어려움이 있다고 한다. 따라서 본 연구에서는 이들 프로그래밍의 어려움을 참조하여 수업모형을 설계하고자 한다. 새로운 프로그래밍언어의 어려움에 대한 문제, 알고리즘의 이해문제, 프로그래밍이 어렵다는 인식 등 여러 방법에서의 문제점을 분석하여 수업모형을 설계한다.

### 3. 실험 및 결과 분석

#### 3.1 앱인벤터를 활용한 코딩교육 수업모형설계

교육용 프로그래밍언어는 자신의 전공이나 전문영역에서 추구해야 할 가치들을 효과적으로 단시간 내에 창출할 수 있는 개발도구이므로 비전공 학습자에게 맞는 수업모형을 개발하여 교육하면 소프트웨어 융합인재를 양성할 수 있다고 본다. 먼저 수업은 정규수업과 비교과 프로그램으로 나누어 진행하였다. 비전공자들의 코딩교육을 위한 수업모형은 [Fig. 1]과 같이 단계별로 설정하여 진행 하였다. 먼저 Jenkins(2002)[17]의 프로그래밍의 어려움 7가지 영역 중에서 처음으로 접하는 프로그래밍 언어의 어려움, 알고리즘의 이해문제, 프로그래밍이 어렵다는 인식 등을 고려하여 학습자에게 맞는 학습내용을 설계하여 개발하였다.



[Fig. 1] Teaching model for coding education

정규수업시간에 학습자들에게 앱인벤터의 다양한 컴포넌트와 센서들의 기능을 살펴본 후 매 수업시간에 문제를 분석한 후 알고리즘과 순서도에 따라 간단한 프로젝트를 작성하였다. 수업진행 과정을 단계별로 살펴보면 교수가 문제를 제시한 후 전체적인 알고리즘을 설명한다. 첫째, 문제 관련 앱인벤터 컴포넌트들을 설명한 후 2인 1조가 되어 문제를 해결할 수 있도록 문제를 분석한 후 알고리즘과 순서도를 작성하도록 한다. 둘째, 순서도에 따라 디자인 한 후 코딩작업을 실시하여 문제를 해결한다. 셋째, 문제가 해결이 되지 않는 학습자들 중 한조가 자신들이 작성한 앱을 발표하도록 하여 학습자들과 교수가 서로 의견을 나누어 틀린 부분을 수정할 수 있도록 서로 정보를 공유한다. 발표는 모든 조가 적어도 한번은

발표할 수 있도록 한다. 넷째, 수업시간에 문제를 해결하지 못한 조는 교수자와 별도의 시간을 정해서 보충수업을 통해 해결하거나 시간이 없는 학생들은 페이스북의 그룹을 통해 교수자와 학생이 정보를 공유함으로써 문제를 해결할 수 있도록 도움을 준다. 다섯째, 매 수업시간마다 수업관련 설문조사를 팽풍 앱을 이용하여 실시하여 어떤 문제를 어려워하는지 체크한 결과를 에버노트에 저장하여 최종 수업모형을 개발하는데 자료로 활용한다. 교수는 매주 수업과 관련된 학습자들의 변화를 작성하여 최종 수업모형을 개발하는데 자료로 활용한다. 여섯째, 비교과프로그램과 연계된 최종과제는 실생활과 혹은 전공과 관련된 앱을 작성하도록 한다. 9주-11주, 3주 동안은 플립러닝 방식으로 진행하였다. 앱을 개발하기 위해 필요한 앱기획 및 분석, 개발과 관련된 내용을 수업 전에 강의지원 예.복습 자료실과 페이스북의 앱인벤터 그룹에 올려놓은 후 학습자들이 사전학습을 하도록 진행하였다. 사전학습여부에 대해서는 성적평가에 반영하기 위해 팽풍 앱을 이용하여 퀴즈로 진행을 하였다. 15주 수업 시간에 앱개발 경진대회에 제출할 프로젝트에 대한 발표를 듣고 학습자들이 팽풍 앱을 이용하여 심사를 하도록 한다. 학습자들의 심사결과와 교수자 두 명의 심사결과를 토대로 수상할 팀을 선정한다. 수업모형의 효과성 검증하는 단계로 설문조사, 개인 인터뷰를 실시했다. 개인 인터뷰는 시간이 많이 요구되므로 팀별로 한명씩 임의로 선정하여 인터뷰를 십분 정도 실시하였다.

##### 3.1.1 플립러닝과 비교과 프로그램

수업진행만으로 학습자들의 코딩교육의 효과를 얻어 내기가 힘이 든다. 비교과과프로그램은 SW 중심사회에서 미래를 이끌어 나갈 창의적인 융합인재 육성을 위해 컴퓨팅 사고력기반의 문제해결능력을 키워주고, 창의적인 아이디어를 앱으로 구현하는 능력을 키울 취지에서 진행하였다. 교과목을 수강한 학습자들이 팀을 편성하여 수업시간에 배운 지식을 경진대회를 통해 자신만의 아이디어로 앱을 개발해봄으로써 수업몰입도와 학습성취력 향상을 유도할 수 있다. 경진대회에 학습자들이 참여함으로써 교육의 성과를 거두고 비전공자 학습자들도 창의적인 앱을 개발할 수 있다는 자신감을 가질 수 있으며 창의적인 활동을 촉진할 수 있다. 수업과 연계된 비교과 프로그램은 앱개발 경진대회이므로 플립러닝 수업모형

에 맞추어 진행을 하였다. 앱개발 관련 내용으로 구성된 동영상과 유인물을 학습자들 수준에 맞추어 미리 작성한 후 강의지원과 페이스북 그룹에 올려놓은 후 학습자들이 사전학습을 하도록 진행하였다. 동영상은 10~15분 정도 시간이 소요되는 것으로 앱개발 할 때 필요한 기획과 디자인 시 주의할 점, 코딩, 테스트 및 최종검토에 대한 내용으로 구성했다. 모바일 앱은 정보의 양, 화면배치, 즉 유저 인터페이스(UI)에 더 신경을 써야 한다. 스마트폰 화면에서는 각각의 기능들을 조밀하게 배치해야 하며, 작은 공간에 적절한 기능들을 손으로 움직이기 편하게 배치하는 부분도 고려하여 작성해야 한다. 이와 관련된 내용들은 유인물로 배포하였다.

학습자들의 사전학습여부 파악을 위해 핑퐁 앱을 이용하여 간단한 퀴즈를 내어 정보를 수집하였다. 핑퐁 앱은 클릭커(Clicker) 프로그램으로써 수업 중에 수업내용에 대한 이해도 파악 및 즉각적인 피드백을 위한 질의응답용으로 활용하며 질문에 대한 결과를 학습자들과 함께 토론할 수 있다. 교수자가 핑퐁 앱을 설치하고 에버노트, 구글 혹은 페이스북 계정으로 로그인한 후 수업을 개설 하면 학습자들은 교수자의 모뎀이름으로 접속한다. 학습자들은 문제의 정답을 입력, 선택하거나 그림을 그려 교수자에게 제출하는 참여형 수업이 가능하다. 학습자들의 반응데이터를 에버노트에 저장한 후 저장한 자료를 바탕으로 학습자들에게 맞는 적절한 교수법으로 운영했다. 한주에 75분수업의 두 번 수업진행으로 수업중간에 수업 집중도가 떨어질 즈음에 스마트폰을 이용하여 다양한 퀴즈, 투표, 학습내용 중 이해가 되지 않는 부분, 즉 추가설명과 예제가 필요한 부분 등을 핑퐁 앱으로 파악하여 진행하였다. 이를 통해 학습자들에게 맞는 맞춤형교육이 가능하고 수업집중도와 흥미를 향상시킬 수 있었음을 알 수 있었다. 설문조사에 의하면 핑퐁 앱이 수업집중도에 영향을 미쳤는가의 질문에 97.44% 학생들이 긍정적인 응답을 하였다. <Table 1>에 결과를 나타내었다.

<Table 1> Pingpong Apps effect your concentration of classes?

	Strongly agree	Agree	Neutral	Disagree	Strongly disagree
Frequency	10	15	13	1	0
Percentage	25.64%	38.46%	33.33%	2.56%	0%

3주 동안 실시한 플립러닝 강의에 만족하느냐의 질문에 긍정적인 응답을 한 학습자들은 <Table 2>에 나타난 것처럼 79.49%를 차지하였다. 이는 학습자들이 코딩교육에 어느 정도 익숙해져 있는 상태에서 플립러닝을 진행한 결과라고 볼 수 있다.

<Table 2> Satisfaction with the flipped learning?

	Strongly agree	Agree	Neutral	Disagree	Strongly disagree
Frequency	6	10	15	6	2
Percentage	15.38%	25.64%	38.46%	15.38%	5.13%

학습자들의 수준차이가 어느 정도 있으므로 다양한 프로젝트를 작성할 수 있도록 수업시간에 여러 개의 프로젝트를 소개한 후 개발할 수 있도록 진행하는 것이 바람직할 것 같다. 비교과프로그램으로 진행된 경진대회를 통해 전문적인 프로그래밍 지식이 없는 비전공자들도 자신의 독창적인 아이디어로 다양한 앱을 개발할 수 있다는 자신감을 갖게 해주는 계기가 되었으며 문제해결능력, 자기 주도적 학습능력, 컴퓨팅적 사고력 향상을 기대할 수 있었다.

### 3.2 연구결과 분석

본 연구의 연구대상은 비전공자들을 대상으로 코딩교육을 효율적으로 운영하기 위한 교육과정이 학습자들의 문제해결능력 향상과 코딩교육에 관심을 가지도록 영향을 미치는 지에 대해 알아보기 위해 경상북도 D대학교의 교양컴퓨터 수업을 수강한 대학생들 39명으로 구성하였다. 남학생 21명(53.85%), 여학생 18명(46.15%)으로 1학년이 4명(10.26%), 2학년이 13명(33.33%), 3학년이 15명(38.46%), 4학년이 7명(17.95%)로 구성되었다. 계열별로는 인문계열 6명(15.38%), 자연계열 8명(20.51%), 예술계열 1명(2.56%), 공학계열 17명(43.59%), 사회계열 4명(10.26%), 경상계열 3명(7.69%)로 나타났다. 앱인벤터 관련 설문조사 결과, 앱인벤터에 대해 이전에 알고 있었던가의 질문에 2명(5.13%)이 앱인벤터에 대해서 들어본 적이 있다고 응답했다. 어려운 부분이 있었다면 어떤 부분이 어려웠는가? 질문에 앱 작성하기 전의 전체설계 부분에 6명(15.38%), 디자인 영역은 3명(7.69%), 블록영역, 즉 코딩영역은 27명(69.23%), 어려운 부분 없다고 응답한 학

생은 3명(7.69%)으로 나타났다. 앱인벤터의 컴포넌트 부분에서는 Connectivity 부분과 Storage 컨트롤, Social 컨트롤을 이해하기가 힘들어 했으며 문법 중에서는 배열(리스트) 영역 19명(48.72%), 프로시저 14명(35.9%), 데이터베이스 영역 5명(12.82%), 제어영역 1명(2.56%) 순으로 나타났다. 수업을 통해 배운 분야에 국한하지 않고 다른 곳에 응용할 수 있다고 생각 하는가?의 질문에 매우 그렇다 11명(28.21%), 그렇다 19명(48.72%), 보통이다 8명(20.51%), 그렇지 않다 1명(2.56%)로 대부분이 긍정적인 응답을 했다. 코딩교육을 통해 창의적인 안목을 기를 수 있다고 생각하는가? 의 질문에 매우 그렇다 12명(30.77%), 그렇다 17명(43.59%), 보통이다 8명(20.51%), 그렇지 않다 2명(5.13%)로 응답했다. 코딩교육을 통해 문제해결능력이 향상되었다고 생각하는가의 질문에 대한 응답은 <Table 3>과 같이 긍정적 응답이 94.87%로 나타났다. 경진대회에 제출할 앱을 자기 주도적으로 개발하는 과정에서 나타난 문제점들을 자신의 힘으로 해결하여 최종적인 앱을 개발한 결과인 것 같다.

<Table 3> Problem solving skills improvement whether or not?

	Strongly agree	Agree	Neutral	Disagree	Strongly disagree
Frequency	10	20	7	2	0
Percentage	25.64%	51.28%	17.95%	5.13%	0%

학습자 자신이 직접 앱을 작성할 수 있다는 것이 신기하고 흥미로운 수업이었다고 응답한 학생들이 38명(97.44%)으로 긍정적이었다. 앱개발 경진대회를 통해 코딩능력이 많아 향상되었다고 응답한 학생들이 35명(89.74%)으로 나타났다. 이 수업을 통해 코딩에 대한 관심과 흥미가 증가되었는가의 질문에 대한 결과는 <Table 4>에 나타난 것처럼 36명(92.31%)이 긍정적으로 응답을 했다.

<Table 4> Increased interest in coding education after classes?

	Strongly agree	Agree	Neutral	Disagree	Strongly disagree
Frequency	10	17	9	3	0
Percentage	25.64%	43.59	23.08	7.69	0%

<Table 5>에는 전공과 융합하여 응용할 수 있다고 생각하는 가의 질문에 89.74% 긍정적인 응답을 하였다.

<Table 5> Possibility of an application to merge with majors?

	Strongly agree	Agree	Neutral	Disagree	Strongly disagree
Frequency	13	14	10	3	1
Percentage	33.33%	35.9%	25.64%	7.69%	2.56%

앱인벤터가 자신의 사고력과 창의력 향상에 도움이 되었는가의 질문에 대한 응답결과는 <Table 6>과 같이 94.87%가 긍정적으로 응답한 것으로 나타났다.

<Table 6> Improve thinking and creativity?

	Strongly agree	Agree	Neutral	Disagree	Strongly disagree
Frequency	15	17	5	2	0
Percentage	38.46%	43.59%	12.81%	5.13%	0%

계열별로 학습 성취도에 차이가 있는지 알아보기 위하여 일원분산분석을 실시했다. 실시한 결과 <Table 7>과 같이 F 통계 값이 2.12, 유의확률은 .0093으로서 유의수준 .05에서 계열별로 학습 성취도에 차이가 없는 것으로 나타났다. 비전공자 학습자들에게 맞는 블록으로 쉽게 작성할 수 있는 코딩수업으로 인해 계열과는 상관없이 자신의 수업참여도에 따라 학습 성취도가 달라지는 것으로 나타났다.

<Table 7> One-Way ANOVA results

	SS	df	MS	F	P
Department	399.67	4	99.92	2.12	0.0993
Error	1600.08	34	47.06		0%
Sum	1999.74	38			

개인 인터뷰를 통해 대부분 학습자들이 문제를 분석한 후 알고리즘 작성과 순서도 작성하는 부분이 처음 접하는 영역이라 힘들었다고 응답했다. 모든 영역이 처음 접하는 부분이라 힘들었지만 시간이 지나면서 어느 정도 이해속도도 빨라졌으며 앱개발에 관심이 가고 수업이 흥미로워졌다고 응답했다. 앱개발은 컴퓨터전공자들만 할

수 있다고 생각했는데 비전공자인 학습자들이 앱을 개발할 수 있음에 자부심을 갖고 전공영역과 융합하여 컴퓨팅 사고력 기반으로 문제들을 해결하려는 관점으로 인식의 변화가 조금은 생겼다고 응답했다. 경진대회 앱을 분류해보면 총 28개 팀이 참여하였다. 관광 앱, 본교 홍보, 소개, 본교 학생식당별 메뉴소개와 찾아가기 앱, 게임 앱, 학교주변 지역관련 앱, 실생활 관련 앱, 먹거리 관련 앱 등이었다. 한 학기 습득한 지식을 바탕으로 다양한 앱을 작성해 봄으로써 학습자들은 코딩교육에 관심을 가지게 되었고 문제해결능력, 사고력, 창의력 등이 향상된 것으로 나타났다. 자신의 전공과 관련된 앱을 개발한 학습자들도 있어서 향후 전공영역과 융합하여 문제를 해결할 수 있는 능력을 향상시킬 수 있을 것으로 기대된다. 코딩교육의 효과에 대한 인식은 문제해결능력 향상 22명(56.41%), 사고력 향상 10명(25.64%), 창의력 향상 7명(17.95%) 순으로 응답하였다.

#### 4. 결론 및 향후과제

교육용 프로그래밍언어는 자신의 전공이나 전문영역에서 추구해야 할 가치들을 효과적으로 단시간 내에 창출할 수 있는 개발도구이므로 비전공자들에게 맞는 수업모형을 개발하여 소프트웨어 융합인재를 양성하여 취업경쟁력을 높일 수 있다고 본다. 비전공자들의 소프트웨어교육을 위한 수업모형 개발이 절실히 요구되는 시점에 본 연구는 비전공자들을 위한 코딩교육 학습모형을 제시하고, 개발한 학습모형이 비전공자들의 전공영역과 융합하여 문제를 해결할 수 있는 능력을 향상시킬 수 있는지 효과를 검증하는데 연구의 궁극적 목적이 있다. 이처럼 비전공자들 대상으로 코딩교육이 절실히 요구되는 시점에 본 연구는 보편적 교육으로서의 코딩교육을 위해 블록형언어의 하나인 앱인벤티를 활용한 교육과정을 제시하였다.

수업진행만으로 학습자들의 코딩교육의 효과를 얻어 내기가 어려우므로 수업과 연계된 비교과프로그램인 앱개발 경진대회를 실시하였다. 자기 주도적 학습을 가능하게 하는 학습자중심의 수업환경으로 진행하기 위해 플립러닝 방식으로 진행을 하였다. 플립러닝 방식으로 진행된 경진대회를 통해 전문적인 프로그래밍 지식이 없는

비전공자들도 자신의 독창적인 아이디어로 다양한 앱을 개발할 수 있다는 자신감을 갖게 해주는 계기가 되었으며 자신의 전공과 연계된 앱개발에도 관심을 가질 수 있는 동기를 부여할 수 있었다. 독창적인 앱을 설계, 분석 및 개발함으로써 문제해결능력, 자기 주도적 학습능력, 컴퓨팅 사고력 향상을 기대할 수 있었다. 경진대회를 마친 후 학생들에게 설문조사를 실시한 결과 스마트한 앱개발 경진대회를 통해 학습자들의 문제해결능력이 향상되었다고 94.87%가 긍정적인 응답을 했으며 전공과 융합하여 응용할 수 있다고 89.74%가 긍정적인 응답을 했다. 이 수업을 통해 코딩에 대한 관심과 흥미가 증가되었다고 92.31%가 긍정적인 응답을 했다. 개인인터뷰를 통해 비전공자인 학습자들이 앱을 개발할 수 있음에 자부심을 갖고 전공영역과 융합하여 문제들을 해결하려는 관점으로 인식의 변화가 조금은 생겼다고 응답을 하였다.

한 학기 습득한 지식을 바탕으로 다양한 앱을 개발해 봄으로써 학습자들은 코딩교육에 관심을 가지게 되었고 문제해결능력, 사고력, 창의력 등이 향상된 것으로 나타났다. 자신의 전공과 융합한 앱을 작성한 학습자들도 있어서 향후 전공영역과 융합하여 문제를 해결할 수 있는 능력을 향상시킬 수 있을 것으로 기대된다. 계열별로 학습 성취도에 차이가 있는지 알아보기 위하여 일원분산분석을 실시한 결과 계열별로 학습 성취도에 차이가 없는 것으로 나타났다. 비전공자 학습자들에게 맞는 블록으로 쉽게 작성할 수 있는 코딩수업으로 인해 계열과는 상관없이 자신의 수업참여도에 따라 학습 성취도가 달라지는 것 같다. 향후 3주 동안 진행한 플립러닝을 10주 이상 진행하여 수업의 효과 차이를 검증해 보고 사물인터넷과 연계한 수업모형 개발을 과제로 남겨둔다.

#### ACKNOWLEDGMENTS

This research was supported by a 2015 Research Grant from Catholic University of Daegu.

#### REFERENCES

- [1] H. S. Yang, I. O. Jeon, "Research on Software Classification

- System based on an Integrated Software Industry”, Journal of Digital Convergence, Vol. 11, No. 4, pp. 91-99, 2013.
- [2] H. S. Woo, M. R. Yeom, D. Y. Jung, “An Analysis on the UCC Media for STEAM Integrated Education”, Journal of the Korea Convergence Society, Vol. 7, No. 1, pp. 43-48, 2016.
- [3] J. Y. Jang, K. H. Lee, S. H. Yun, “A Study on Convergence Curriculum Development for Idea Creation”, Journal of Digital Convergence, Vol. 12, No. 2, pp. 571-576, 2014.
- [4] Y. S. Son, K. J. Lee, “Computational Thinking Teaching Model Design for Activating IT Convergence Education”, Journal of the KIECS, Vol. 11, No. 5, pp. 511-522, 2016.
- [5] K. M. Kim, H. S. Kim, “A Case Study on Necessity of Computer Programming for Interdisciplinary Education”, Journal of Digital Convergence, Vol. 12, No. 11, pp. 339-348, 2014.
- [6] 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.
- [7] S. H. Kim, S. K. Han, H. C. Kim, “A Study on Learner’s Characteristics and Programming Skill in Computational Literacy Education”, The Journal of Korean Association of Computer Education, Vol. 13, No. 2, pp. 15-23, 2010.
- [8] K. G. Kim, “Development and Application of Software Education Program based on Computational Thinking”, Kyungpook National University, Master’s thesis, 2015.
- [9] S. J. An, “Direction and Prospects of Software Education”, [http://edzine.kedi.re.kr/2015/autumn/article/policy\\_01.jsp](http://edzine.kedi.re.kr/2015/autumn/article/policy_01.jsp), 2015.
- [10] S. H. Kim, “Analysis of Non-Computer Majors’ Difficulties in Computational Thinking Education”, The Journal of Korean Association of Computer Education, Vol. 18, No. 3, pp. 15-23, 2015.
- [11] H. S. Lee, S. H. Kang, C. S. Kim, “Analysis of the Structural Relationship among Factors Related to the Effects of Flipped Learning”, The Journal of Korean Association of Computer Education, Vol. 20, No. 1, pp. 87-100, 2016.
- [12] S. G. An, Y. M. Seo, Y. J. Lee, “A Review and Synthesis of Research in Educational Programming Language”, The Korea Society of Computer Information Winter Conference, Vol. 20, No. 1, pp. 139-142, 2012.
- [13] D. J. Kim, E. Y. Ha, “The Future Direction of Information Education in University according to Computerization”, Journal of Digital Convergence, Vol. 13, No. 10, pp. 33-40, 2015.
- [14] S. H. Park, “The Effectiveness of Learning Community for the Development of Convergence of University Students”, Journal of Digital Convergence, Vol. 13, No. 9, pp. 29-37, 2015.
- [15] S. Yu, J. Kang, “Correlation between Critical Thinking Ability and Disposition of University Nursing Students for the Development of Convergence”, Journal of Digital Convergence, Vol. 13, No. 6, pp. 197-203, 2015.
- [16] B. H. Kim, “Computer Programming Education using App Inventor for Android”, Journal of the Korea Institute of Information and Communication Engineering, Vol. 17, No. 2, pp. 467-472, 2013.
- [17] T. Jenkins, “On The Difficulty of Learning to Program”, 3rd Annual LTSN-ICS Conference, Loughborough University, pp. 53-58, 2002.

피수영(Pi, Su Young)



- 2000년 8월 : 대구가톨릭대학교 전산통계학과(이학박사)
- 2012년 3월 ~ 현재 : 대구가톨릭대학교 교양교육원 조교수
- 관심분야 : 소셜마케팅, 스마트교육, IT융합, 교육콘텐츠
- E-Mail : syupi@cu.ac.kr