



Journal of Knowledge Information Technology and Systems

ISSN 1975-7700 (Print), ISSN 2734-0570 (Online)

<http://www.kkits.or.kr>

Application of Goal-based Scenario for Liberal Arts Programming Education

Young-Taek Jin*

Department of computer Engineering, Hanbat National University

ABSTRACT

New technologies such as big data analysis and artificial intelligence are evolving based on convergence with information technology and are affecting all industries. To cultivate problem-solving skills in line with this trend, universities are providing programming education as a basic liberal art course. However, students who take programming courses as liberal arts are having a lot of difficulties in connecting the problem-solving process with actual programming. Even at the stage of putting together basic programming elements and applying them to problem-solving, they tend to feel that they can't do anything or give up. To solve this problem, this paper proposes a method of applying a goal-based scenario to programming education. The goal-based scenario is one of the case-based learning approaches such as problem-based learning and action learning. Based on the scenarios, students can clearly understand the context/situation in which particular knowledge is used and learn to apply the earned knowledge to solving specific problems. Therefore, programming learning based on these scenarios can be used as a tool that can be applied to solving and analyzing similar problems, away from syntax and result-oriented education. In this study, we analyze the existing Python liberal arts programming education and present the effects and considerations of the method of applying the goal-based scenario.

© 2020 KKITS All rights reserved

KEYWORDS: Problem-solving skills, Programming education, Case-based learning, Goal-based scenario, Python programming, Syntax-oriented learning

ARTICLE INFO: Received 19 October 2020, Revised 23 November 2020, Accepted 11 December 2020.

*Corresponding author is with the Department of Computer Engineering, Hanbat National University, 125

Dongseo-daero, Yuseong-gu, Daejeon 34158, KOREA.
E-mail address: ytjin@hanbat.ac.kr

1. 서론

인공지능(AI), 블록체인, 사물인터넷 (IoT) 등 소프트웨어 기술이 4차 산업혁명을 선도하는 중심 기술이 되고 있다. 전 산업분야가 소프트웨어와의 융합을 통해서 디지털화가 가속화되면 코딩 역량은 창의적인 문제해결능력을 위한 토대로서 점점 더 필수역량이 되어 갈 것이다.

이런 추세에 발맞추어 대학에서는 기초 교양으로서 프로그래밍 교육을 제공하고 있다. 또한 프로그래밍이 문제해결에 대한 새로운 관점을 제공하기 때문에 모든 계열의 학생에게 학습을 권장하고 있다. 그러나 학생, 특히 교양으로서 프로그래밍 교과목을 수강하게 되는 학생들은 프로그래밍 구문과 작성방법을 학습하고 나서도 문제해결 과정을 실제 프로그래밍과 연결하는데 많은 어려움을 겪고 있는 실정이다. 프로그래밍 학습에서 기본 문법을 배워 기본적인 프로그래밍을 하는 단계는 성취를 많이 느끼는 단계이고 가장 흥미가 많이 유발되는 단계이다. 그러나 문법 지식을 종합해서 문제해결에 적용하는 단계에서는 아무것도 할 수 없다고 느끼거나 포기하는 경향을 보이고 있기도 하다. 이런 문제를 해결하기 위하여, 프로그래밍 교육에 목표 중심 시나리오(goal-based scenario)를 적용한 방법을 제시하고자 한다. 목표중심시나리오란 학습자들이 해결해야 할 과제를 시나리오의 형태로 제공하고, 이를 해결하기 위해 기술과 관련 지식을 사용하여 학습목표를 달성하게 되는 교수-학습방법을 의미한다. 문법중심의 교육에서 벗어나 이런 시나리오를 토대로 하는 프로그래밍 학습은 자연스럽게 프로그래밍을 유도하고, 이전 문제의 해결 경험을 유사문제의 해결과 분석에 적용할 수 있는 도구로 활용될 수 있다.

본 연구에서는 기존에 적용된 파이썬 교양 프로그래밍 수업방식을 분석하고 시나리오기반 학습을

적용한 방법의 효과와 고려사항 등을 제시하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 프로그래밍 학습의 어려움과 이런 어려움을 극복하기 위한 다양한 기존 학습 방법을 분석한다. 3장에서는 목표중심시나리오에 대해 서술하고, 시나리오를 활용한 프로그래밍 교육 방법을 제안하고 적용 결과를 분석한다. 4장에서는 결론 및 시사점을 서술한다.

2. 프로그래밍 교육 방법

2.1 프로그래밍의 어려움

프로그래밍은 학습하기 쉽지 않은 기술이며, 기본적인 프로그래밍 요소와 문법을 학습하고 난후에 이런 지식을 종합해서 완성된 프로그램을 작성하는 단계에 이르기까지는 많은 어려움이 발생하고 있다[1]. 프로그래밍 학습의 어려움과 해결방법 연구에서 제시한 몇몇 프로그래밍 어려움들은 다음과 같이 요약할 수 있다[2].

- 프로그래밍 언어는 전문적인 사용을 위해 개발되었고 높은 수준의 추상화가 필요하다.

- 프로그래밍 언어는 암기해야 할 복잡한 구문 세부 사항이 있고 학습자들은 구문 규칙에 집중하는 동시에 알고리즘 구성에 집중해야 하는 것을 요구한다.

- 프로그래밍 어려움은 특정 오류 및 오해에 의해 야기된다.

프로그래밍과 관련한 어려움을 패턴으로 제시한 연구[3]에서는 학습자, 교수자 관점에서 마주치는 어려움에 관한 데이터를 수집하고 블룸(Bloom)의 6가지 단계별로 패턴을 제시하였다. 학습자의 관점에서 가장 어려운 점은 변수와 관련된 내용 및 구문 오류라고 제시하고 있다.

이와 더불어 컴퓨터 전공자가 아닌 비전공 학습자들이 기초 프로그래밍에서 겪는 어려움과 이를 해결하기 위한 방안을 제시하는 연구[4]도 있다. 비전공자들이 컴퓨팅 사고 교육에서 어려워하는 부분은 명령어의 기능과 응용, 아이디어를 어떻게 어떤 코드로 작성해야 하는지에 대한 어려움이라고 밝히고 있다. 또한 프로그래밍은 높은 수준의 인지 부하를 수반하게 되고 이런 어려움은 프로그래밍에 대한 경험이 없고 코딩 과정에서 부딪치는 다양한 오류에 기인한다고 밝히고 있다[5].

2.2 교양 프로그래밍 학습모델

한국교육학술정보원의 소프트웨어 교육 교수학습 모형 개발 연구에서는 5가지 소프트웨어 교수학습 모델을 제시하였다. 컴퓨팅 사고력중심의 소프트웨어 학습 모델로서는 분해, 패턴인식, 추상화, 알고리즘 및 프로그래밍(P) 항목을 포함한 DPAAP 모델이 제시되었다[6]. 분해(Decomposition)는 해결 가능한 단위로 문제를 분해하는 것을 의미하며, 패턴인식(Pattern Recognition)은 반복되는 일정한 경향 및 규칙의 탐색을 의미한다.

추상화(Abstraction)는 문제의 단순화 및 패턴인식으로 발견한 원리의 공식화를 의미한다. 알고리즘(Algorithm)은 핵심 원리를 절차적으로 구성하는 것이며 프로그래밍(Programming)은 프로그래밍 언어로 구현하고 실행하는 것을 의미한다.

컴퓨터 비전공자들을 위한 소프트웨어 교육모델로서 파이썬(python) 기반 소프트웨어 교육 모델과 학습 내용을 제시하는 연구[7]가 있다. 제시한 모델은 위에서 언급한 5가지 소프트웨어 교수학습 모델 중에서 개발중심(DDD)모델을 토대로 파이썬 프로그래밍 교육에 적용하고 이에 대한 효과를 제시하였다.

비전공자의 컴퓨팅사고를 함양하기 위한 교육모

델로서 문제기반학습(PBL)을 적용한 다양한 연구가 있다. 비전공자의 IT 융합 교육을 위한 PBL 활용 교육 모델 연구[8]에서는 컴퓨팅 사고를 배양하기 위해 문제기반학습을 적용한 결과를 제시하고 있다. 이 연구에서는 실제 프로그램은 작성되지 않았고 사후 설문분석 결과를 통해 문제기반학습 방법이 컴퓨팅 사고에 많은 도움이 된다는 것을 보여주고 있다. 문제기반학습을 스크래치 프로그래밍에 적용한 연구[9]에서는 스크래치 교육을 위한 교육내용 및 설문도구를 개발했다. 수업에 적용한 결과, 컴퓨팅 사고력과 문제 해결 능력이 향상되었음을 제시하고 있다. 소프트웨어 교육의 학습효과를 높이기 위해 프로그래밍 학습 자료에 컴퓨팅 사고를 적용하여 기존 강의법을 개선하는 시도도 있다[10]. 또한 스토리텔링과 프로그래밍을 융합한 교육방법[11]과 플립러닝 모델을 활용하여 프로그래밍 수업에 적용하는 방법도 제시되었다[12]. 복잡한 문제의 해결능력을 배양하기 위해 코딩 교육을 적용하는 연구가 있다[13]. 이 연구에서는 순차, 선택, 반복으로 이루어지는 알고리즘을 통해서 문제가 해결되는 방법을 제시하고 있다.

기존 연구를 분석해볼 때 대학에서 소프트웨어 교육은 크게 두 가지 방향으로 구분할 수 있다 [14-16]. 첫째, 교양으로 비전공자를 대상으로 하는 경우 창의력과 문제해결력 향상을 중점으로 교육하는 방향이 있다. 둘째, 문제 해결력을 배양하기 위해 코딩 능력과 기술을 배양하기 위한 교육 방향이 있다. 드물게는 컴퓨팅사고 교과목과 프로그래밍 언어를 학기로 연계해서 진행하는 방향이 있다. <표 1>에 제시된 바와 같이 컴퓨팅사고와 프로그래밍 학습은 서로 연계되고 보완되는 관계를 갖는다. 따라서 두 가지 교육방향을 균형있게 진행하는 것이 바람직하나 제한된 시간과 환경 및 전문화된 코딩 교육의 어려움으로 인해 코딩보다는 문제 해결을 위한 컴퓨팅 사고의 배양에 중점을 두는

경향이 많다. 그러나 프로그래밍이 없이 문제 해결책을 알고리즘으로 작성하는 경우도 알고리즘 표현을 위한 의사코드(pseudo code) 또는 유사한 수단이 여전히 필요하다. 특히 그 해결책에 대한 검증을 위해 컴퓨터를 활용할 필요가 있다. 따라서 종합적인 관점에서 볼 때, 프로그래밍 능력은 필요하며 기본 프로그래밍 기술을 습득할 수 있는 기회는 교양 교과목을 통해서 제공될 수 있다. 코딩 능력을 배양하기 위해서는 적절한 프로그래밍 교수 방법이 요구된다. 본 연구에서는 기존 연구에서도 출된 문제를 해결하기 위해 목표중심시나리오를 활용한 프로그래밍 교수 방법을 제시하고자 한다.

표1. 프로그래밍과 컴퓨팅 사고와의 관계
Table1. The relationship between programming and computational thinking

프로그래밍	컴퓨팅 사고	내용
요구 분석 능력: 문제의 이해와 분석	문제 분해	문제의 해결을 용이하기 위해 큰 문제를 작은 문제들로 분해
	패턴 찾기	문제에서 반복되는 사항이나 규칙의 탐색
설계 능력: 자료 구조의 선택과 구체적인 알고리즘의 작성	추상화	문제 해결에 필요한 근본적이고 일반적인 요소에 집중
	알고리즘 작성	문제 해결을 위해 적용해야 할 근본 해결책을 체계적으로 기술
구현 능력: 알고리즘을 프로그래밍 언어로 표현	프로그래밍	특정한 프로그래밍 언어 구문(입력, 처리, 출력)을 활용하여 해결책을 작성

3. 시나리오 기반 프로그래밍 교육

3.1 목표중심시나리오

목표중심시나리오(goal-based scenario)는 R.

Schank를 중심으로 미국 Northwestern 대학의 연구자들에 의해 1980년대 초에 구체화된 학습설계 이론이다[17]. 다양한 분야에서 목표중심시나리오를 활용한 연구가 많이 제시되어 있다[18,19].

목표중심시나리오란 5가지 유형의 사례기반학습의 하나로써 학습자들이 해결해야 할 과제를 시나리오의 형태로 제공하고, 이를 해결하기 위해 스킬 (skills) 및 관련 지식을 사용하여 학습목표를 달성하게 되는 교수-학습방법을 의미한다[20].

목표중심시나리오를 활용한 학습 특징 중의 하나는 학습을 새로운 문제해결을 위해 사례들을 축적하는 과정으로 본다는 것이다. 문제를 해결하려는 과정에서 기대 실패(expectation failure)가 발생할 수 있다. 기대 실패는 잘될 것으로 기대했는데 실패하는 경우를 말한다. 이때의 실패는 긍정적인 방향으로의 실패도 포함한다. 즉, 잘 되기 어려운 것으로 기대했는데 기대 이상의 결과를 얻게 되는 경우도 기대 실패가 일어난 것이며, 왜 이런 일이 일어났는지를 분석하는 과정에서 학습의 기회를 얻게 된다. 프로그래밍 학습의 경우 이런 경험을 많이 하게 되며, 따라서 문제 해결책을 찾고 해결책을 실행하기 위해 축적된 지식과 경험을 활용하게 된다. 문제가 해결된 후에는 그 해결 사례가 축적되어 추후에 유사한 문제가 생겼을 때 보다 효과적인 방법을 찾는데 기여하게 된다.

목표중심시나리오는 7가지 요소로 구성되어 있다.

(1) 목표: 목표는 시나리오를 통하여 학습자들이 획득하기를 원하는 기술을 말한다.

(2)임무: 임무는 학습자들이 설정된 목표를 성취하기 위하여 수행해야 하는 과제이다. 임무는 학습자들이 임무를 계속해서 추구할 수 있도록 동기를 유발시킬 수 있는 것이어야 하며, 따라서 임무는 학습자들이 처해 있는 실제 상황과 유사한 형태의 과제가 제시되는 것이 좋다.

(3) 커버 스토리: 커버 스토리는 목표 달성을 위하여 학습자들이 수행할 임무와 관련한 맥락을 이야기 방식으로 설명함으로써 임무수행에 필요한 내용을 구체화시킨 것이다. 이를 통하여 임무를 정확히 이해하고 임무 수행을 위하여 무엇을 해야 할지를 학습자들이 알게 된다.

(4) 역할: 역할은 학습자들이 표지 이야기 속에서 맡게 되는 역할이다.

(5) 시나리오 활동: 시나리오 활동은 학습자들이 임무를 수행하는 모든 구체적인 활동을 의미한다. 설정한 목표나 임무에 따라 다양한 시나리오 활동이 존재하며 의사결정을 하거나 지시를 내리는 것, 질문에 답하는 것, 도구를 사용하고, 필요한 정보를 검색하는 것 등이 포함된다. 프로그래밍의 경우, 임무를 달성하기 위한 단계별 프로그래밍 과정이 제시되는 중요한 활동이다.

(6) 학습 자원: 자원은 학습자들이 임무를 수행할 때 필요로 하거나 참고해야 할 내용을 말한다. 이러한 정보는 학습자원의 형태로 잘 조직되어 있어야 하며 어렵지 않게 접근할 수 있도록 준비되어야 한다.

(7) 피드백: 학습자들이 임무를 수행해 가는 과정에서 발생할 수 있는 어려움을 해결하는데 제공되는 교수자의 도움을 의미하며, 적절한 시기에 제공되어서 학습자들이 그 피드백을 이용할 수 있어야 한다.

3.2 시나리오의 설계

프로그래밍은 다양한 측면에서 수행하기 쉽지 않은 지적 활동이다. 교수법 측면에서 교수자는 프로그래밍 언어를 사용하여 문제 해결을 촉진시키기 보다는 프로그래밍 언어 자체 및 구문 세부 정보를 가르치는데 더 집중하는 측면이 있다. 학습자인 학생의 관점에서는 과거 문제 해결 경험에 대

한 유추를 못하고 이전 지식을 새로운 문제에 적용하지 못한다. 또한 가능한 해결책을 빨리 찾지 못하면 종종 문제 해결을 포기하거나 문제를 완전히 이해하지 않은 채 문제를 해결하려고 한다[3]. 이러한 문제를 해결하기 위해 목표중심시나리오를 설계하여 코딩 교육에 적용한 예는 <표 2>와 같다.

표2. 주문처리시스템을 위한 목표중심시나리오 사례
Table 2. An example of the goal-based scenario for order processing system

시나리오 1~N	
목표	주문처리 시스템의 개발
임무	1)주문 총액의 계산하기 2)주문량에 따라 배송료 계산 3)고객의 유형에 따라 할인을 계산하기 4)다양한 할인정책을 수립하기
커버스토리	쇼핑몰 창업자로서 고객의 상품주문에 따른 처리를 원한다
역할	개발자로서 요구분석, 설계, 구현(프로그래밍)을 수행한다
시나리오 활동	1)필요한 자료(입력,출력)의 식별 및 표현 2)처리조건 분석: 조건의 구체적인 의미와 표현(프로그램 구문) 3)해결책(알고리즘)의 표현 및 고려사항
학습자원	교재의 내용과 사례: 자료구조(리스트, 튜플,파일), 조건(if문), 반복(while, for문), 클래스(class)
피드백	오류의 이해, 코드로의 표현, 코드 변경, 코드의 비교

목표는 한 학기에 걸쳐 학습한 프로그래밍 지식을 종합해서 구축하고자 하는 목표 시스템이다. 시나리오의 임무는 프로그래밍 구문 및 문법을 점차로 배우게 됨에 따라 수행해야 하는 일을 의미하며 난이도가 증가하도록 설계했다. 이전 문제해결에 사용된 지식을 다음 문제에 활용할 있도록 구조적 유사성(문제를 해결하는 원리가 유사한 것을

의미, 구조 유사성이 높을수록 전달이 잘 일어남)을 보장하기 위해 하나의 시스템을 점차로 확장해 나가면서도 통합된 하나의 시스템으로 일관되고 서로 유기적으로 연결될 수 있도록 설정하였다. <표 2>에서 제시된 바와 같이 임무는 주문처리 시스템 문맥 내에서 수행할 수 있도록 시나리오1에서는 주문총액 계산하기, 시나리오2에서는 배송료 계산, 시나리오 3에서는 고객유형에 따른 할인을 계산하기 등등 N개의 시나리오 마다 다양한 임무가 있다. 이러한 임무는 프로그래밍 구분지식과 적용경험이 증가할수록 난이도가 증가한다.

3.3 시나리오의 적용 및 결과

목표중심시나리오를 활용한 프로그래밍 교육은 2019년도 1,2학기 및 2020년도 1학기에 교양으로 개설된 파이썬 프로그래밍 수업에 적용하였다. 대상은 공학계열 학과의 2019학년도 신입생(1학기:39명, 2학기:29명) 및 2020학년도 신입생(1학기:34명)을 대상으로 이루어졌다. 기존 수업방식은 2019학년도 신입생에게 적용되었고, 목표중심시나리오 방식은 2020학년도 신입생에게 적용되었다. 수업은 이론2시간, 실습2시간의 형태로 이론 내용은 각각 “기본 문법 알아보기(숫자, 변수, 입,출력)”, “데이터구조(문자열, 리스트, 튜플, 딕셔너리)”, “조건문(if, elif) 및 반복문(for, while)”, “함수 사용하기”, “예외 및 파일처리”, “모듈과 패키지 사용하기”, “클래스 작성하기”로 구성되었다. 이론 수업 방식은 파이썬 프로그램의 구문(문법)설명을 하고 난후 간단한 적용사례를 제시하고 예제 프로그램을 설명하는 방식으로 이루어졌다. 실습은 이론 시간에 제시된 단편적인 사례를 실제로 수행해보는 방식으로 진행되었으며 실습 후에 학생들은 <표 3>과 같은 실습성찰일지를 작성하여 제출하였다.

표3. 실습성찰일지의 사례
Table 3. The example of Lab. reficetion journal

점검 항목	(아니다) X	(그렇다) O
프로그램의 수행시 오류 없는 동작 및 해결		
문제에 대한 파이썬 코드의 이해		
유사한 문제에 적용하기 위한 코드 변경		
대안 프로그램의 장,단점 및 고려사항의 이해		
실습에서 가장 어려웠던 부분 및 해결노력의 서술 (오류, 해결과정의 문제점 등을 서술)		

피드백은 실습성찰일지의 내용을 분석하여 다음 수업에서 결과를 설명하는 형태로 이루어졌다. 전체적으로 기존 교육 방법에서는 프로그램이 작성되어 가는 단계별 과정이 제시하기 보다는 이미 작성되어 있는 예제 프로그램을 설명하는 형태의 정적인 강의방식이 채택되었다.

이러한 기존 프로그래밍 수업을 분석한 결과는 <표 4>에 4가지 항목으로 제시되었다. 실습은 학기별로 12주에 걸쳐 수행이 되었고 실습단위를 3주 단위로 그룹해서 요약했다.

“프로그램의 수행시 오류없는 동작 및 해결” 항목의 경우 실습 초반이라 어렵지 않은 내용이라 오류없이 실습을 수행한 비율이 대략 평균적으로 70%에 이르고 있다.

‘문제에 대한 파이썬 코드의 이해’ 항목의 경우, 학습 교재에 제시된 코드를 약 68%의 학생이 이해하고 있다. 그러나 “유사한 문제에 적용하기 위한 코드 변경”과 “대안 코드의 장, 단점 및 고려사항의 이해”항목의 경우, 실습 초반과 후반에 관계없이 충족 비율이 낮다는 것을 알 수 있다.

표4. 파이썬 프로그래밍 실습점검표의 통계 요약
 Table 4 The statistics summary of python programming
 Laboratory reflection journal

(단위:%)	실습 (1~3)		실습 (4~6)		실습 (7~9)		실습 (10~12)	
	미충족	충족	미충족	충족	미충족	충족	미충족	충족
프로그램의 수행시 오류 없는 동작 및 해결	26	74	29	71	28	72	35	65
시나리오의 적용	23	77	21	79	27	73	38	62
문제에 대한 파이썬 코드의 이해	25	75	49	51	25	75	31	69
시나리오의 적용	26	74	39	61	22	78	30	70
유사한 문제에 적용하기 위한 코드 변경	51	49	69	31	72	28	69	31
시나리오의 적용	45	55	59	41	53	47	49	51
대안 코드의 장,단점 및 고려사항의 이해	75	25	71	29	66	34	85	15
시나리오의 적용	65	35	52	48	60	40	75	35

이 분석결과가 시사하는 바는 기존에 작성된 프로그램은 이해하고 있으나 유사한 문제에 이전 지식을 적용하지 못하고 있고, 동일 문제와 비교가 될 수 있는 코드가 제시되지 않았기 때문에 어떻게 작성하는 것이 좋은지 장점과 단점을 파악할 수 없다는 것을 의미한다.

이런 문제점을 해결하기 위해 시나리오의 적용에서 본 연구에서는 크게 세 가지 부분에 중점을 두었다.

1) 일관되고 확장 가능한 임무

임무는 설정된 목표를 달성하기 위해 수행되는 과제이다. 임무는 프로그래밍 문법에 대한 이해와

응용능력의 향상에 맞추어 확장될 수 있는 형태로 부여했다. 이전 임무와 무관한 단편적인 연관이 없는 개별 사례로 제공되는 것이 아니라 목표를 달성하기 위해 이전 지식을 활용할 수 있도록 동일한 문맥 안에서 제공하였다.

2) 시나리오 활동

학습자들이 프로그래밍에서 어렵다고 느끼는 부분 중의 하나가 프로그램이 작성되는 단계별 과정이 제시되지 않는다는 점이다. 강의 자료에는 문제에 대한 분석과 결과 프로그램 예제가 제시되지만 단계별 프로그래밍 과정은 제시되지 않아서 학습자들이 프로그래밍 과정을 학습하기 어려워한다. 따라서 <표 2>에서 제시된 시나리오 활동에 따라서 단계별 프로그래밍 과정을 제시하였다. 학생에게 과정을 직접 보여주어야 했기 때문에 진행과정의 코드를 입력할 때 소규모 프로그램이라도 천천히 강의할 수밖에 없었다. 필요한 자료(입력 및 출력)의 식별 및 변수 명명에서 부터 적절한 자료구조의 선택과정이 제시되었다. 그 다음에는 처리조건의 분석을 진행하였다. 조건의 구체적인 의미가 분석되고 학습한 프로그램 구문에서 후보 구문을 선택하는 과정이 제시되었다. 마지막으로 프로그램 구문을 정리하고 프로그램을 실행하는 과정이 제시되었다. 프로그램을 작성하는 과정에서 실수가 생각보다 많이 발생했기 때문에 학생은 실수를 진단하고 수정하는 방법을 알게 되는 측면이 있지만 준비하는 입장에서는 실수를 줄이고 신속한 복구를 위해 준비를 많이 해야만 했다. 중간크기 이상의 프로그래밍 과정을 보여줄 때는 뼈대 코드를 만들어 놓은 다음에 이것을 단계별로 채워가며 설명하는 방식으로 진행했다.

3) 피드백

학생이 작성한 실습성찰일지를 분석해서 이전 수업에서 가장 어려웠던 점을 설명하는 피드백을

제공하였다.

이런 중점사항을 반영하여 목표중심시나리오를 적용한 결과는 <표 4>의 '시나리오 적용'항목에 제시되었다.

시나리오를 적용한 결과는 약간의 효과를 제시하고 있지만 적용기간이 한학기이고 대상이 30명에 불과해서 통계적으로 유의미하다고 할 수는 없다. 또한 여기서 주의할 점은 비교를 위해서는 동일집단 내에서 통제그룹과 그렇지 않은 그룹을 비교해서 비교해야 하지만 <표 4>에 제시된 내용은 2019 1,2학기 및 2020 1학기 수강생이 각각 다르므로 결과 또한 각 대상에 대해서 분석한 내용이므로 직접적으로 비교할 수 없다는 점이다. 이러한 주의에도 불구하고 설문과 실습성찰 일지를 분석한 결과, "유사한 문제에 적용하기 위한 코드 변경"과 "대안 프로그램의 장, 단점 및 고려사항의 이해" 측면에서 개선이 있다는 점이다. 이런 향상은 이전 문제해결 경험을 시나리오 형태로 제시하여 기존 경험을 새로운 문제에 체계적으로 적용했기 때문이다. 특히, 임무를 수행하기 위한 단계별 시나리오 활동이 직접적으로 제시되었고, 임무들은 단일 문맥에서 유기적으로 제시되었으며 자료의 분석과 처리조건의 분석이 프로그램 구문과 대응해서 제시되었기 때문이다.

4. 결 론

소프트웨어와 컴퓨터는 경제, 사회, 인문 등 모든 분야와 긴밀히 연관되어 있고 복잡한 문제를 컴퓨터를 활용하여 창의적으로 해결하기 위한 프로그래밍 능력이 요구되고 있다. 또한 프로그래밍이 문제해결에 대한 새로운 관점을 제공하기 때문에 대학에서는 비전공인 인문사회 계열의 학생에게 학습을 권장하고 있다. 그러나 교양으로서 프로그래밍 교과목을 수강하게 되는 학생들은 문제해결 방법과 과정을 실제 프로그래밍과 연결하는데

많은 어려움을 겪고 있는 실정이다. 이런 문제를 해결하기 위하여, 본 논문에서는 프로그래밍 교육에 목표중심시나리오를 적용한 방법과 그 결과를 제시하였다. 이 방법의 적용 및 결과 분석을 통해서 시사하는 점은 다음과 같다. 첫째, 설문분석을 통해서 파악했듯이 학생들이 프로그래밍에서 어렵다고 느끼는 점은 자신의 문제해결 아이디어와 알고리즘을 어떻게 프로그램으로 표현하는지 모른다는 점이다. 이러한 문제를 해결하기 위해 시나리오에 제시된 활동을 단계별로 직접 보여줌으로써 해결가능하다. 둘째, 단편적인 코드가 아니라 목표 시스템을 서브 시스템으로 나누고 일관된 맥락에서 제시함으로써 다양한 시나리오를 통해 기존 해결과정을 새로운 문제 시나리오에 적용할 수 있다. 마지막으로 피드백 과정을 통해서 오류 해결과 코드 비교를 수행함으로써 프로그래밍을 복습하는 기회를 가질 수 있다. 향후 본 연구의 확장방안으로 쉽게 이용할 수 있는 워크북 형태의 체계적인 시나리오를 작성한다면 효과적인 프로그램 학습이 이루어질 것이다.

References

- [1] T. Erik, <https://www.thinkful.com/blog/why-learning-to-code-is-so-damn-hard>, Sep. 2020.
- [2] A. Gomes, and A. J. Mendes, *Learning to program - difficulties and solutions*, International Conference on Engineering Education, pp. 283-287, 2007.
- [3] Y. Bosse, and M. A. Gerosa, *Why is programming so difficult to learn?: Patterns of difficulties related to programming learning mid-stage*, ACM SIGSOFT Software Engineering Notes, Vol. 41, No.

- 6, 2016.
- [4] S-W. Kim, *Analysis of non-computer majors' difficulties in computational thinking education*, The journal of korean association of computer education, Vol. 18, No. 3, pp. 49-57, 2015.
- [5] Y-H. Shin, H-J. Jung, and J-S. Song, *Analysis of learning experience in design thinking-based coding education for sw non-major college students*, Journal of Digital Contents Society, Vol. 20, No. 4, pp. 759-768, 2019.
- [6] KERIS, *A research on the development of teaching and learning models for sw education*, Report CR 2015-35, 2015.
- [7] 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.
- [8] E-S. Kang, S-I. Shin, and K-J. Lee, *Education model using pbl for IT convergence education of non-major in liberal arts class: focusing on computing thinking*, Journal of Digital Contents Society, Vol. 20, No. 11, pp. 2159-2168, 2019.
- [9] Y-S. Han, *Effectiveness of problem-based learning based programming education : focus on computational thinking*, Asia-pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology Vol. 8, No. 7, pp. 433-445, 2018.
- [10] J-H. Park, and D-W. Park, *Application of computational thinking to improve learning effectiveness in the subject 'programming'*, Journal of Knowledge Information Technology and Systems, Vol. 13, No. 2, pp. 263-271, Apr. 2018.
- [11] J-M. Koo, *A development of convergent teaching method using storytelling and programming*, Journal of Knowledge Information Technology and Systems, Vol. 11, No. 6, pp. 713-723, Dec. 2016.
- [12] S-J. Kim, and D-E. Cho, *A study on learning model for effective coding education*, Journal of the Korea Convergence Society, Vol. 9, Issue 2, pp. 7-12,2018.
- [13] K-S. Song, Y-J. Ko, and I-S. Jeon, *Study on the analysis system for complex problem Solving skill in the area of coding education environment*, The Journal of Korean Institute of Information Technology, Vol. 17, No. 11, pp.105-114, 2019.
- [14] J-E. Nah, *Software education needs analysis in liberal arts*, Korean Journal of General Education, Vol. 11, No. 3, pp. 63-89, 2017.
- [15] W-S. Kim, *Exploring the direction of granular basic-software education considering the major of college students*, Journal of The Korean Association of Information Education, Vol. 23, No. 4, pp. 329-341, 2019.
- [16] G-J. Park, and Y-J. Cho, *Exploratory study on the direction of software education for the non-major undergraduate students*, Journal of Education & culture of Inha university, Vol. 24, No. 4, pp. 273-292, 2018.
- [17] R. C. Schank. *The design of goal-based scenarios*, The Journal of the Learning Sciences, Vol. 3, No. 4, 1993.
- [18] D. Hung, D-T. Chen and S. C. Tan, *A social-constructivist adaptation of case-based reasoning: Integrating goal based scenarios with computer-supported collaborative*

learning, Educational Technology, Vol. 43, No. 2, pp. 30-35, 2003.

- [19] K-L. Cho, Y-H. Cho, M-K. Kim, and B-S. Sung, *Design and development of wbi contents : a goal-based ccenarios (gbs) model approach*, The Journal of Korean Association of Computer Education, Vol. 7, No. 5, pp. 9-21, 2004.
- [20] Jonassen, *Typology of case based learning: the content, form and functions of cases*, educational Technology, Vol. 46, No. 4, pp. 11-15, 2006.

교양 프로그래밍 교육을 위한 목표중심 시나리오의 적용

진영택

한밭대학교 컴퓨터공학과 교수

요 약

빅 데이터 분석, 인공지능 등과 같은 분야의 새로운 기술은 정보기술의 융합을 토대로 발전하고 있으며 모든 산업분야에 영향을 끼치고 있다. 이러한 추세에 부합하는 문제해결 능력을 배양하기 위해 대학에서는 기초 교양으로서 프로그래밍 교육을 제공하고 있다. 그러나 교양으로서 프로그래밍 교과목을 수강하게 되는 학생들은 문제해결 과정을 실제 프로그래밍과 연결하는데 많은 어려움을 겪고 있는 실정이다. 또한 프로그래밍 기본 요소를 종합해서 문제해결에 적용하는 단계에서는 아무것도 할 수 없다고 느끼거나 포기하는 경향을 보이고 있다. 이러한 문제를 해결하기 위하여, 본 논문에서는 프로그래밍 교육에 목표중심 시나리오를 적용한 방법을 제시하고자 한다. 목표중심 시나리오는 문제기반 학습 및 액션러닝과 같은 사례기반 학습 접근 방식 중 하나이다. 시나리오를 바탕으로 학생들은 특정 지식이 사용되는 상황을 명확하게 이해하고 습득한 지식을 특정 문제 해결에 적용하는 방법을 배울 수 있다. 따라서 이런 시나리오를 토대로 하는 프로그래밍 학습은 구문 및 결과중심 교

육을 벗어나 유사문제의 해결과 분석에 적용할 수 있는 도구로 활용될 수 있다. 본 연구에서는 기존에 적용된 파이썬 교양 프로그래밍 수업방식을 분석하고 목표중심시나리오를 적용한 방법의 효과와 고려사항 등을 제시하였다.



Young Taek Jin received the bachelor's degree in the Department of Computer Engineering from the Chung-Ang University in 1981. He received the M.S. degree and the Ph.D. degree in the Department of Computer Engineering from Chung-Ang University in 1983 and 1992, respectively. From 1983 to 1990, he was a researcher at The Korea Institute of Energy Research. He has been a professor in the Department of Computer Engineering at Hanbat National University since 1990. His current research interests include software product line system for micro architecture, programming education methods. He is a regular member of the KKITS.

E-mail address: ytjin@hanbat.ac.kr