

대학 프로그래밍 입문 수업에서 짝 프로그래밍의 효과: 학습성과와 학생 만족도 측면에서

정충교

강원대학교 컴퓨터학부

요 약

대학교 프로그래밍 입문 수업에 짝 프로그래밍 방식의 실습을 도입하여 운영한 경험과 교훈을 소개한다. 짝 프로그래밍을 적용하지 않은 반과 적용한 반의 수업 성과와 설문조사 데이터를 통해 짝 프로그래밍의 효과를 분석하였다. 그 결과 상당 수준의 짝 프로그래밍 성과가 관찰되었지만 짝 프로그래밍 방식으로 실습한 학생 전반적으로 효과가 있다고 보기는 어려웠다. 일부 학생들에게서는 뚜렷한 학습 성과가 있었고 일부 학생들은 그렇지 못하였다. 짝을 어떻게 지어 주는가가 수업 성과와 학생 만족도에 영향을 주는 매우 중요한 요소임이 설문조사를 통해 드러났다. 이 수업 운영 경험을 바탕으로 짝 짓는 방안을 포함하여 짝 프로그래밍을 수업에 적용하는 방법을 제시한다.

키워드 : 짝 프로그래밍, 프로그래밍 교육, 대학, 입문 과정

Effects of Pair Programming in an Introductory Programming Course for College Students: Academic Performance and Student Satisfaction

Choongkyo Jeong

Dept. of Computer, Kangwon National University

ABSTRACT

This paper presents the results of adopting pair programming practices in an introductory university programming course. The effectiveness of pair programming is analyzed by comparing the performance and survey data gathered from the experimental class that adopted the practice to those from the control class that did not. An analysis of the performance data found that pair programming improved student performance; however, not all students in the pair programming class benefited from the practice as only some students showed clear improvements in learning outcomes. The survey showed that the specific method for matching students into pairs is a critical factor affecting their performance and satisfaction. Based on the lessons learned from this pair programming class, I propose some suggestions and guidelines for applying pair programming to programming classes including 'pair matching'.

Keywords : Pair programming, Programming education, College, Introductory class

2016년도 강원대학교 대학회계 학술연구조성비로 연구하였음 (관리번호-520160199).

논문투고 : 2017-08-28

논문심사 : 2017-09-04

심사완료 : 2017-09-27

1. 서론

짝 프로그래밍은 두 사람이 짝을 이뤄 함께 프로그램을 작성하는 방법이다. 두 사람이 한 대의 컴퓨터를 이용하며, 한 사람이 운전자(driver) 역할을 맡아 키보드와 마우스를 차지한 상태로 코드를 작성하고, 다른 사람은 항해사(navigator) 역할을 맡아 알고리즘을 생각하고 작성 중인 코드에 문제가 없는지 관찰한다. 운전자 역할과 항해사 역할은 일정 시간 간격(10분~20분)으로 서로 바꿈으로써 공동작업성을 높인다.

짝 프로그래밍은 1990년대 초반부터 문헌을 통해 소개되기 시작했는데 1999년 Kent Beck이 극한 프로그래밍(eXtreme Programming) 개발 방법론[7]을 제안하면서 짝 프로그래밍을 실행방법들(practices) 중 하나로 포함시켰고 그 이후로 짝 프로그래밍이 널리 알려지게 되었다. 극한 프로그래밍은 기존의 소프트웨어 개발 방법론에서 추구하던 여러 목표와 수단들을 극한 수준까지 끌어 올리고자 하는 시도이다. 소프트웨어 품질을 높이기 위해 널리 받아들여지는 수단들 중 하나가 바로 코드 리뷰이다. 코드 리뷰는 한 사람이 작성한 코드를 다른 사람이 검토함으로써 코드의 오류를 줄이고 구조를 개선하는 활동인데 이 코드 리뷰를 극단적인 수준으로(실시간으로) 수행하는 것이 바로 짝 프로그래밍이라고 볼 수 있다.

산업 현장에서 짝 프로그래밍을 올바른 방식으로 적용하는 경우 아래와 같은 여러 가지 유익한 효과가 있는 것으로 보고되었다[1].

- 두 사람이 개별적으로 코드를 작성할 때에 비해 개발 시간은 조금 늘어나지만 코드 오류가 줄어든다. 이는 프로젝트 성격에 따라 이득일 수도 있고 손해일 수도 있을 것이다.
- 개발자들이 더 즐거운 마음으로 일을 하게 된다. 이는 물론 두 사람이 잘 맞는 경우이며 개인 성격과도 관련이 있다.
- 소프트웨어 설계 품질이 우수하다. 두 사람의 토론을 통해 더 짧고 효율적인 코드를 작성하게 된다.
- 학습 효과가 높다. 두 프로그래머가 서로 다른 지식과 시각을 나눔으로써 새로운 것을 배우게 된다.
- 프로그래머 간 소통이 증진되며 팀웍이 좋아진다.

특히 짝을 바꾸어 가면서 개발을 진행하는 경우 팀 내 프로젝트 공유도가 높아진다.

그러나 짝 프로그래밍이 항상 좋은 효과를 내는 것은 아니며 조건이 맞는 경우에만 유익하다고 알려져 있다. 문제가 지나치게 단순한 경우에는 짝 프로그래밍을 통해 이득을 얻을 수 없으며, 프로그래머의 수준이 매우 높은 경우에도 각각 작업하는 편이 더 낫은 것으로 보고되었다[15]. 또, 문제 자체가 프로그래머에게 익숙한 문제일 경우 짝 프로그래밍의 이득을 얻을 수 없다는 연구 결과도 있다[9]. 짝 프로그래밍이 알려진 것과 달리 일반적으로는 효율과 생산성 측면에서 이득이 있다고 말할 수 없다고 주장하는 연구 결과도 있다[2]. 그러나 이 연구에서도 복잡한 과제의 경우와 학습을 위한 경우에는 뚜렷이 긍정적인 효과가 있다고 말하고 있다.

이 논문은 2016학년도 2학기에 강원대학교 컴퓨터학부에 개설된 "자바프로그래밍" 과목[16]에 짝 프로그래밍을 도입하여 운영한 경험을 제시하고 짝 프로그램의 효과를 분석한다. 이 과목(자바프로그래밍)은 주로 컴퓨터학부의 컴퓨터정보통신공학전공을 이수하는 1학년 학생들이 수강하는데 수강생 중에는 1학년이 아닌 재수강 학생들도 포함되어 있다. 이 과목은 1학기에 개설되는 "프로그래밍기초" 과목의 후속 과목으로서 수강생들은 이미 "프로그래밍기초" 과목을 통해 알고리즘 개념, 프로그램 흐름 제어, 변수, 배열, 함수 선언과 호출, 테스트와 디버깅 등 프로그래밍의 기초 개념을 알고 있으며 기본적인 실기 경험을 갖고 있는 상태이다. "자바프로그래밍" 과목에서는 "프로그래밍기초" 과목에서 학습한 내용을 바탕으로 객체지향프로그래밍 기법을 배운다. 객체의 개념을 익힌 후 컬렉션과 인터페이스를 주로 사용하여 견고하면서도 확장성 있는 구조를 갖도록 설계하고 리팩토링하는 연습을 한다.

이 과목은 짝 프로그래밍을 도입하기에 적절한 과목이다. 짝 프로그래밍의 성과에 관한 일반적인 연구[15]에서 밝혀진 것처럼 문제가 지나치게 단순한 경우에는 짝 프로그래밍 자체가 그다지 의미가 없다. 협업의 가치가 없기 때문이다. 따라서 학생들이 프로그래밍을 처음 접하고 기초적인 문법과 도구 등을 배우는 "프로그래밍기초" 과목보다는 어느 정도의 난이도를 갖는 문제를 풀게 되는 "자바프로그래밍" 과목이 더 적합하다고 생

각한다. 또, 고학년에서 배우는 특정 기술 분야에 특화된 과목의 경우에는 분야의 특수성 때문에 일반성이라는 측면에서 연구 대상으로서의 가치가 떨어진다고 생각한다.

2. 관련 연구

학교 교실에서의 짝 프로그래밍에 대한 연구는 교육 효과, 학생 만족도, 짝 선정 방식에 따른 효과, 짝 프로그래밍 수용에 대한 학생의 태도 등 다양한 측면에 걸쳐 이루어 졌다. 특히 서양에서는 학교 프로그래밍 수업에서의 짝 프로그래밍 효용에 대한 연구가 매우 풍부하게 이루어 졌다[6][10][12][13]. Norsaramah 등은 일정 수준을 만족하는 74개의 연구 논문들을 선정하고 분석하여 학교 실습실에서의 짝 프로그래밍 효용에 대한 연구 결과를 종합적으로 정리하였다[11]. 그 내용을 요약하면 아래와 같다.

- 짝 프로그래밍을 하는 경우 혼자 하는 경우에 비해 대체적으로 학생들이 과제를 더 빨리 완료한다.
- 짝 프로그래밍을 하는 경우 혼자 하는 경우에 비해 대체적으로 시험과 실습과제에서 더 나은 성적을 올렸다. 그러나 일부 반대의 결과가 나온 경우도 있어 짝 프로그래밍의 이득이 확실하다고 말하기는 어렵다.
- 짝 프로그래밍을 하는 경우 학생들의 수업에 대한 만족도가 높으며 수업을 더 즐겁게 느낀다.

우리나라에서는 교실 프로그래밍 수업에서의 짝 프로그래밍 도입 효과에 대한 연구가 많지 않다. 그 연구들도 비교적 제한된 범위 내에서만 이루어진 것으로 보인다. (아래에서 간단히 둘러 볼 것이다.) 우리나라는 문화적으로 서양과 많이 다르고 특히 학교 수업 분위기는 서양 대학과 큰 차이가 있어 우리나라 학교 수업에 짝 프로그래밍을 도입할 때의 효과에 대한 관찰과 연구가 필요하다.

국내 교실에서의 짝 프로그래밍 효과를 살펴본 여러 연구들 중 일부[3][5][8]는 비교적 단순한 문제를 다루는 전문계 고등학교의 기초수업을 대상으로 수행했다는 점에서 아쉬운 부분이 있다. 서론에서 언급했듯이 지나치

게 단순한 문제를 다룰 때는 짝 프로그래밍의 효과 낮은 것으로 알려져 있기 때문이다. 김정량은 초등학생들에게 짝 프로그래밍을 적용한 결과를 보고하였다[4]. 초등학교와 전문계 고등학교 학생들을 대상으로 국내 교실에서의 짝 프로그래밍 효과를 살펴본 이 연구들은 대체로 짝 프로그래밍이 학업성취도, 자기효능감, 수업만족도 등에서 긍정적 효과가 있다고 보고하고 있다. 전소은은 대학교 1학년 C언어 프로그래밍 기초 과목에 짝 프로그래밍을 적용한 연구 결과를 발표했다[14]. 그러나 실험 및 관찰 대상 과목이 프로그래밍을 처음 접하는 학생들을 위한 도입 과목이며, 짝 프로그래밍 효과를 측정하기 위해 학생들에게 주어진 과제가 비교적 단순하였으며 측정 시간 및 횟수도 매우 제한적이었다.

국내에 IT 관련 학과가 설치된 대학이 많아 대학에 개설되는 입문 프로그래밍 과목이 매우 많은데 반해 짝 프로그래밍을 도입하여 운영하고 효과를 관찰한 연구 보고가 거의 없다. 이는 짝 프로그래밍에 대한 국내외의 많은 연구들의 결과가 현장에서 인정되지 않고 있거나, 짝 프로그래밍을 이용한 교육에 대한 정보가 보급되지 않았기 때문일 것이다. 이 연구는 대학의 입문 프로그래밍 과목에 짝 프로그래밍을 실제로 도입하여 적용한 경험을 정리하여 제시한다.

3. 짝 프로그래밍 적용 수업 운영

나는 2016학년도 2학기에 강원대학교 컴퓨터학부에 개설된 "자바프로그래밍" 과목의 두 개 분반을 담당했다. 이 과목은 내가 지난 10년 동안 맡아 온 과목이다. 15주에 걸쳐 진행되는 3학점 실습 과목이며 매 주 이론 강의 2시간, 실습 2시간 수업을 한다. 나는 일방적인 이론 강의의 효과가 낮은 것으로 판단하여 2013년부터는 이론 수업과 실습 수업을 모두 실습실에서 실습 위주로 진행하면서 짧은 여러 개의 강의를 실습 사이 사이에 삽입하는 방식으로 수업을 진행해 왔다. 대체로 강의 시간의 합은 한 시간 정도이고 나머지 세 시간은 실습과질의 응답에 사용했다. 그 동안 강의 내용과 실습 과제는 꾸준히 보완해 왔지만 짝 프로그래밍을 도입한 것은 2016년 강좌가 처음이었다.

중간고사 이전까지는 두 반 모두 짝 프로그래밍을 적

용하지 않았다. 객체지향프로그래밍 개념을 익히는 초보적인 내용에서는 짝 프로그래밍이 크게 의미가 있지 않다고 생각했다. 또한 두 반반 학생들의 학습 능력 수준을 인위적으로 동등하게 맞출 수 없는 상황에서 짝 프로그래밍의 효과를 관찰하기 위한 실험 계획에 의한 것이기도 했다. 중간고사 이후부터 두 반 중 한 반에만 짝 프로그래밍을 도입하였다. 지금부터는 짝 프로그래밍을 적용한 반을 "짝반", 적용하지 않은 반을 "홀로반"이라고 부르겠다.

중간고사 후 첫 수업에서 짝반 학생들에게 짝 프로그래밍에 대해 설명해 주고 앞으로의 계획을 말해 주었다. 짝 프로그래밍의 유래, 목적, 방법, 효과 등을 설명하고 협조를 당부했다. 실습 효과 제고, 공동작업 연습, 공동체 조성 등이 핵심 목표로 제시되었다. 그리고 그 이후 기말고사까지 진행한 6주간의 수업에서 매주 약 3시간 가량의 실습(강의한 시간 별도)을 짝 프로그래밍 방식으로 진행했다. 매주 수업에서 주제와 목표가 제시되고 간단한 설명과 예를 통한 시범을 보인 후 학생들이 짝을 이루어 주어진 실습을 수행하도록 했다. 실습 결과는 매주 채점하여 돌려줌으로써 스스로 수업 성과를 진단하게 하였다.

실습과제의 난이도와 분량은 예년과 비슷하게 제시했는데 이는 평균 수준의 학생들이 수업 외에 약 세 시간 내지 다섯 시간 정도 추가로 노력해야 완수할 수 있는 정도이다. 짝반 학생들은 실습시간에 다 하지 못한 과제들을 수업 후에 실습실에서 짝과 함께 하거나 집에서 온라인 협동작업 프로그램을 이용해 짝 프로그래밍 방식으로 마무리하도록 권했으나 수업 외의 시간에 이것이 잘 지켜지는지 세세하게 관리할 수는 없었다.

짝반 학생들은 모두가 짝을 지어 실습을 하도록 했다. 짝 프로그래밍의 성과에 가장 큰 영향을 주는 요소는 문제의 성격과 짝 조합인 것으로 알려져 있다[11]. 실력이 아주 뛰어난 두 사람이 짝을 이루는 경우 별다른 성과를 얻지 못하며, 반대로 실력이 매우 모자라는 두 사람이 짝을 짓는 경우에도 효과가 거의 없다고 한다. 짝을 이루는 두 사람의 실력이 비슷한 경우에 좋은 효과를 얻을 수 있고, 실력이 다소 차이가 있는 경우에도 이득이 있다. 실력 차이가 다소 나는 경우, 상대적으로 하수의 질문을 통해 새로운 관점으로 문제를 보는 기회를 얻게 되고, 상대적 하수는 고수로부터 효율적으로 지식을 받아들일게 된다. 이 수업에서는 학생들을 중간고사 성적을 기준으로 정렬하고 1/3 가량 등위 차

이가 나도록 짝을 지어 주었다. 짝은 2주 동안 유지되도록 하고 그 기간이 지나면 사람을 바꾸어 새로이 짝을 지어 주었다. 그리하여 중간고사부터 기말고사까지 기간 동안 학생들은 서로 다른 세 사람과 짝을 맺었다. 짝을 바꾸어 주면 잘 맞지 않는 짝 조합으로부터 발생하는 불만을 완화할 수 있으며 학생들의 교류를 촉진하고 여러 사람과의 협업 경험 기회를 제공하는 효과가 있다.

짝 프로그래밍을 하는 학생들은 두 사람이 한 대의 컴퓨터만을 이용해 코드를 작성하도록 했으며 15분 간격으로 운전사, 항해사 역할을 바꾸도록 했다. 획일적으로 시간을 맞춰 역할을 바꾸도록 하면 흐름이 끊어질 수 있으므로 시간만 알려주고 가급적 그 시간 전후로 교대하도록 하였다. 그러나 실제로는 이를 잘 따르지 않는 짝도 적지 않아 반복적, 지속적으로 협조 요청을 해야 했다.

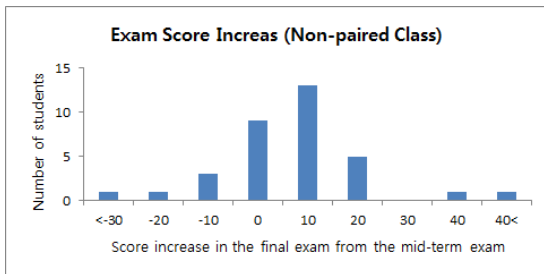
4. 성과 측정

기말고사 시험을 통해 학습성취도를 측정하고 설문조사를 이용해 학생들의 주관적 평가를 조사하였다. 중간고사와 기말고사는 각각 한 시간 동안의 필기시험과 세 시간 동안의 실기시험으로 나누어 치렀다. 필기시험에서는 아무 자료도 참고할 수 없으나 실기시험에서는 수업 중에 부과된 실습과제의 모범답안을 포함하여 모든 자료를 볼 수 있게 하였다. 짝반과 홀로반이 같은 문제로 같은 시간에 시험을 치렀다.

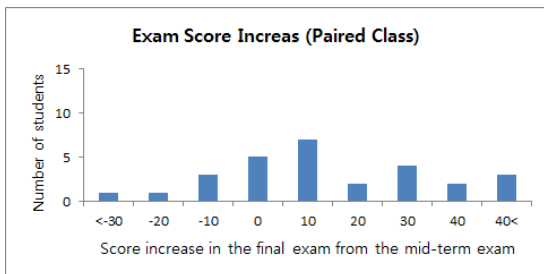
두 반 학생들의 학습 능력과 기초실력에 차이가 있을 수 밖에 없으므로 두 반의 성적을 단순하게 비교를 하는 대신 두 반 각 학생의 중간고사 성적 대비 기말고사 성적 향상치를 관찰하여 학습성취도 측면에서의 짝 프로그래밍 성과를 측정했다. 각 학생의 필기시험과 실기시험을 합한 중간고사, 기말고사 점수 각각을 100점 만점으로 환산한 후 중간고사 대비 기말고사 점수 상승을 계산했다. 홀로반의 경우 점수 상승 평균값이 2.0, 표준편차가 15.4였고, 짝반은 평균값이 7.8, 표준편차가 21.0이었다. (Fig. 1)은 각 반 학생들의 점수 상승을 도수분포 그래프로 그린 것이다. 홀로반 학생은 34명이고 짝반 학생은 28명이다.

점수 상승 평균값만을 보면 홀로반의 상승 평균은 2.0이고 짝반의 상승 평균은 7.8이므로, 홀로반에 비해

짝반의 평균값이 5.8점 높다. 비교적 큰 차이인 것처럼 보이지만 이 차이가 통계적으로 얼마만큼이나 의미를 갖는지 보기 위해 유의성검정을 해 본다. "짝 프로그래밍을 통해 학생들의 (중간고사 대비 기말고사) 점수 상승 평균값이 높아지지 않는다"는 귀무가설을 놓고, 두 표본 t-검정(two-sample t-test)을 수행하면, t 통계량이 1.223이고 p-value가 0.119가 된다. 그러니까 12% 유의수준에서만 귀무가설을 기각할 수 있으며, 통상적인 수준인 10%, 혹은 5% 유의수준에서는 귀무가설을 기각할 수 없다. 중간고사 이후로 짝 프로그래밍을 적용하는 경우 그렇지 않은 경우에 비해 학생들의 기말고사 점수 상승의 평균값이 높아진다고 확신할 수는 없다는 의미이다. 이 실험에서 두 반의 점수 상승 평균값의 차가 5.8로 상당한 차이가 났지만, 각 반의 점수 상승 표준편차 또한 매우 커서 이런 결과가 나왔다. 그러나 평균값만을 살펴본 이 결과가 짝 프로그래밍의 효과를 부정하는 근거가 될 수는 없다.



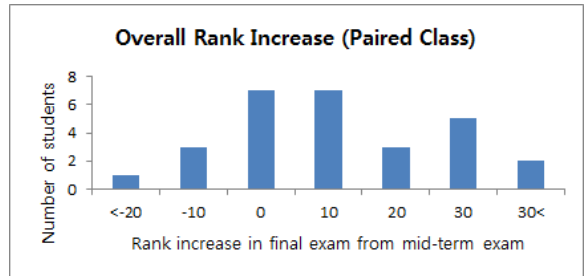
(a) Non-paired class.



(b) Paired class.

(Fig. 1) Score increase in the final exam compared to the mid-term exam:

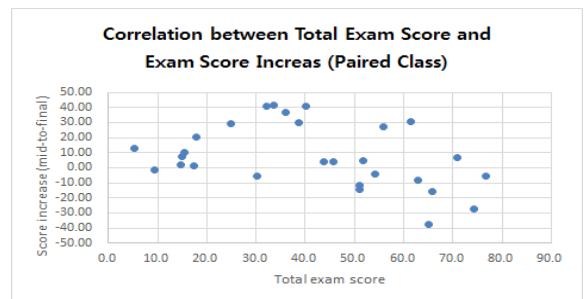
(a) Non-paired class, (b) Paired class.



(Fig. 2) Overall rank increase in the paired class (final exam rank minus mid-term exam rank).

(Fig. 1)의 두 그래프를 비교해 보면 그래프의 왼쪽 부분은 두 그림이 비슷한 반면 오른쪽 부분에서 큰 차이를 보인다. 그러니까 짝 프로그래밍이 학생들 전반에 영향을 주기 보다는 일부 학생들에게만 뚜렷한 효과가 있다는 것이다. 이 부분을 좀 더 면밀하게 살펴봐야 하겠다. (Fig. 2)는 짝반 학생들의 전체 등수 상승을 보여 주는 도수분포그래프이다. 홀로반과 짝반 학생 전체를 중간고사 점수와 기말고사 점수 각각에 따라 등수를 매긴 후, 짝반 학생들의 중간고사 등수 대비 기말고사 등수 상승을 집계한 결과이다. 이 그래프를 통해서도 성적 향상이 뚜렷한 짝반 학생 그룹이 눈에 띈다. 그러니까 효과가 거의 없는 학생들이 있고 뚜렷한 효과가 있는 학생들이 있다는 것이다.

(Fig. 3)은 짝반 학생들의 중간고사 점수와 기말고사 점수를 각 50%씩 반영한 최종 점수와 중간고사 대비 기말고사 점수 상승의 관계를 보여 주는 산포도이다. 대체로 중간 정도 수준의 학생들에게서 높은 점수 상승을 볼 수 있다. 상위권 학생 두 명이 매우 큰 하강을 보였는데 이는 어떤 추세라기보다는 개인적 사정에 의한 것으로 짐작된다.



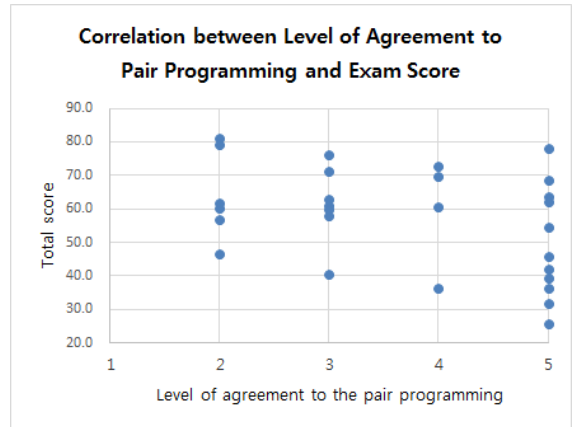
(Fig. 3) Correlation between the total exam score and the exam score increase (Paired class).

<Table 1> Survey questionnaire and mean scores of the replies.

Question	Mean score
Do you agree with the purpose of pair programming?	4.00
Does pair programming help you improve your programming skills?	3.36
Does pair programming help you improve cooperative working skills?	3.75
Does pair programming help create a classroom atmosphere with more active social interactions such as making friends or sharing difficulties?	3.82
Are you in favor of applying pair programming to next year's classes?	3.71

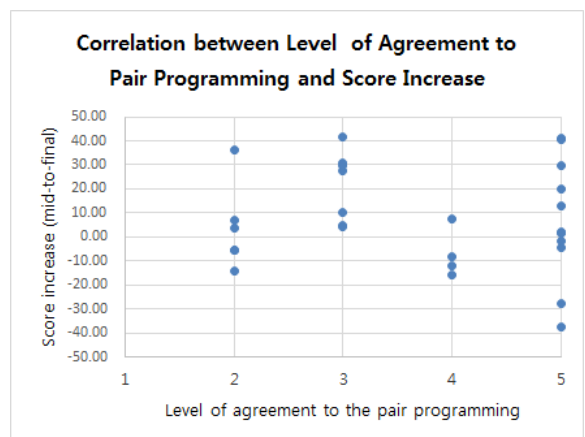
학기 말에는 짝반 학생들을 대상으로 설문조사를 실시했다. 설문조사에서는 짝 프로그래밍의 목적에 공감하는지, 그것이 능력 향상에 도움이 되는지, 협동작업 연습에 도움이 되는지, 공동체 활성화에 도움이 되는지, 그리고 내년에도 짝 프로그래밍을 하는데 찬성하는지를 물어 보고, 각 질문에 대해 1점부터 5점까지 다섯 등급 척도로 답하게 했다. 각 질문에 대한 응답의 평균값은 <Table 1>과 같다. 짝 프로그래밍의 목적에는 대체로 공감하는 편이지만 실제 효과는 기대에 다소 못 미치는 것으로 느꼈다고 볼 수 있다. 또, 프로그래밍 능력을 높이는데 직접적으로 도움이 되는 정도에 비해 협동작업 연습, 친구 사귀기 등 프로그래밍 외적인 요소에서 더 큰 이득이 있는 것으로 느꼈다고 볼 수 있다.

(Fig. 4)는 내년 수업에 짝 프로그래밍을 적용하는 데 대한 찬성 정도와 성적(중간고사와 기말고사를 50%씩 반영한 총점)간 상관관계를 보여 준다. 내년 강좌에 짝 프로그래밍을 적용하는 데 대한 찬성 정도를 짝 프로그래밍에 대한 종합적인 평가로 볼 수 있을 것이다. 짝 프로그래밍에 대한 평가는 학생의 성적과는 뚜렷한 상관관계가 없어 보인다. 다만, 짝 프로그래밍에 대해 좋은 평가를 내린 학생들 중에는 성적이 낮은 학생이 많고, 짝 프로그래밍에 대해 좋지 않은 평가를 내린 학생들 중에는 성적이 좋은 학생이 다소 많아 보인다. 상대적으로 우수한 학생이 그렇지 않은 학생과 함께 하면서 부담을 느낀 반면, 상대적으로 역량이 떨어지는 학생들은 더 잘 하는 짝으로부터 도움이 받는다고 느낀 결과일 것이다.



(Fig. 4) Correlation between the level of agreement with the pair programming pedagogy and exam score.

(Fig. 5)는 짝 프로그래밍을 적용하는 데 대한 찬성 정도와 중간고사 대비 기말고사 성적 상승 간 상관관계를 보여 준다. 이 그림 역시 일관된 상관관계를 나타내지는 않는다. 즉, 짝 프로그래밍 실습 기간 동안 성적 상승이 크게 있었다고 하더라도 그에 비례하여 짝 프로그래밍에 대한 만족도가 높은 것은 아니라는 것이다. 이에 대한 해석은 설문조사의 서술식 답변으로부터 얻을 수 있다. 설문조사에서는 학생들이 자유롭게 소감을 적을 수 있게 했다. 28명의 짝반 학생들 중 16명이 서술 응답을 했는데 그 중 9개의 서술 응답이 짝과의 관계에 대한 언급이었다. 짝과 잘 맞지 않아 어려움이 있다는



(Fig. 5) Correlation between the level of agreement with the pair programming pedagogy and the exam score increase.

<Table 2> The gathered data summary.

Class	Year	Number of students (a)		Number of students excluded from the analysis (b)				Number of students included in the analysis (a-b)		Experimental unit
				Students who failed to get credit due to too many absences		Students who have not taken one or more of lab test, written test in midterm and final exams				
		Number	Partial Sum	Number	Partial Sum	Number	Partial Sum	Number	Partial Sum	
Singles	Freshmen	17	39	0	1	3	4	14	34	Control group
	Retaking	22		1		1		20		
Pairs	Freshmen	32	39	1	2	9	9	22	28	Treatment group
	Retaking	7		1		0		6		
Sum	Freshmen	49	78	1	3	12	13	36	62	
	Retaking	29		2		1		26		

것이다. 그러니까 실력과 성격에서 짝과 잘 맞지 않으면 짝 프로그래밍을 통한 성과와 별개로 짝 프로그래밍에 대한 만족도가 떨어진다고 생각할 수 있다.

<Table 2>는 이 연구에 사용된 데이터 전체와 정제 내역을 보여 준다. 두 반의 수강생 수는 78명인데 이 중 62명의 데이터만 분석에 사용되었다. 총 수업 시간 중 1/3 이상 결석하여 학칙에 따라 F 학점을 받은 학생들과 중간고사, 기말고사의 각 필기시험이나 실기시험을 보지 않은 학생들은 분석에서 제외하였다. 분석에 사용된 62명의 데이터에는 출석 기준을 만족하고 중간고사, 기말고사를 제대로 치렀으나 점수가 낮아 F 학점을 받은 7명의 데이터가 포함되어 있다.

이 수업에서 다룬 구체적인 학습 내용에 대해서는 자세한 설명을 생략한다. 이 수업에서 사용된 15주 동안의 강의자료와 실습과제는 강의 홈페이지[16]에서 볼 수 있다. 중간고사와 기말고사 필기시험과 실기시험 문제와 모범답안도 공개되어 있다.

5. 검토 및 결론

데이터를 통해 짝반이 홀로반에 비해 평균 5.8점 더 높은 성적 상승(중간고사 대비 기말고사 성적 상승)을

거둔 것으로 나타났다. 이는 작지 않은 수치이지만 표준 편차가 큰 탓에 상승의 평균값이 갖는 의미는 다소 제한적이라고 할 수 있다. 또 짝 프로그래밍의 효과를 보지 못하는 학생들과 상당한 효과를 보는 학생들로 나뉘는 사실을 데이터를 통해 관찰할 수 있다. 그러니까 효과를 보는 학생들을 위해서는 적극적으로 효과를 더욱 높이는 방안을 강구하고 그렇지 않은 학생들에게는 짝 프로그래밍을 하지 않게 하거나 아니면 효과가 나타나지 않는 이유를 밝혀 이를 해소해 주어야 할 것이다. 그러나 수집된 데이터만으로는 효과를 보는 학생들이 갖는 공통적 특징을 파악하기 어려웠다. 상위권 학생들이 대체로 큰 효과를 거두지 못한다는 점 정도만 확인할 수 있다. 또한, 짝 프로그래밍을 통해 성적 상승을 이루었는지 여부가 짝 프로그래밍에 대한 만족도와 큰 상관성이 없다는 점도 확인하였다. 짝 프로그래밍에 대한 만족도는 성적 상승보다는 짝과의 조화가 큰 영향을 준다는 점을 서술식 설문 답변을 통해 확인하였다.

이 경험을 통해 나는 짝 프로그래밍의 유용성을 보았다고 여긴다. 홀로반과 짝반의 수업 분위기는 사뭇 달랐다. 홀로반의 경우 실습에 들어 가면 교실이 조용하고 질문이 주로 강사와 조교에게 집중된다. 짝반은 강사와 조교에게 거의 질문을 하지 않는다. 대신 실습 시작 후 시간이 조금 지나면 짝지어 문제를 해결하기 위해 교실

전체가 술렁이고 소란해 진다.

문제는 강사가 짝 프로그래밍을 어떻게 운영하는가에 달려 있는 것 같다. 특히 짝을 어떻게 지어주는가 하는 것이 매우 중요해 보인다. 상위권 학생들 중 희망하는 학생들은 혼자서 실습을 할 수 있도록 허용해 주고, 나머지는 비슷한 수준의 학생들을 짝지어 주되 하위권 학생들로 만들어진 짝에게는 강사와 조교가 집중적으로, 적극적으로 개입하여 지도해 주는 방식이 바람직할 것으로 생각한다. 그리고 짝이 잘 맞지 않는 경우에는 적절한 방법으로 이를 신속히 파악하고 해결해 주는 방안을 마련하는 것도 꼭 필요할 것이다.

교실 짝 프로그래밍에서 운전자와 항해사로서의 역할 구분이 엄격할 필요는 없는 것 같다. 두 사람이 짝을 지어 하나의 코드를 완성해 나가면서 서로 토론하고 협업하는 작업 자체에 더 큰 의미가 있다고 생각한다. 설문 조사에서도 학생들이 짝 프로그래밍의 기술적 성과에 비해 협업과 교류 측면에 더 높은 만족을 느끼는 것으로 나타났다. 짝 프로그래밍을 통해 만들어진 공동학습 환경이 각 학생의 학습 동기를 전반적으로 상승시키고 이것이 성적 상승으로 연결되는 것으로 해석할 수 있다.

그럼에도 불구하고 적당한 시간 간격으로 두 사람이 운전자와 항해사 역할을 바꾸는 것 자체는 꼭 필요하다고 본다. 그렇게 해야 어느 한 사람이 소외되는 것을 방지할 수 있기 때문이다. 한 사람이 코딩하는 동안 다른 사람이 멍 하니 앉아 있거나 스마트폰을 사용하는 등 다른 일을 한다면 좋은 효과를 기대할 수 없다. 학생들이 짝 프로그래밍에 더 충실하게 임하도록 유도하는 반강제적인 방법으로서 각 학생이 자신의 짝에 대해 짝 프로그래밍에 충실하게 임하는 정도를 평가하게 하고 이를 적절히 성적에 반영하는 방안도 생각해 볼 수 있다. 이 방법을 적용하려면 짝을 여러 번 바꾸어 줌으로써 여러 사람으로부터 평가를 받도록 해야 한다. 그렇게 해야 어느 한 사람의 의도적인 혹은 비의도적인 부당한 평가에 의한 왜곡을 완화할 수 있다.

매년 매우 많은 프로그래밍 입문 강좌들이 대학에 개설되고 있고 강좌를 담당할 강사들은 효과적인 교육을 위해 노력하고 있다. 그러나 그런 노력에도 불구하고 많은 학생들이 어려움을 호소하고 그 중 일부는 좌절하곤 한다. 우리나라 학생들의 경우 그 어려움이 더 크지 않을까 생각한다. 온라인과 오프라인 모두에서 친절하고 2

상세하게 씌어진 한글 자료가 영어 자료에 비해 매우 부족하기 때문이다. 초중고 과정에서 자료를 찾아 보며 스스로 문제를 해결하는 연습이 되어 있지 않다는 것도 우리 학생들의 약점이다.

짝 프로그래밍은 다른 사람과 토론하고 협조하면서 문제를 함께 해결해 나감으로써 우리 학생들이 이런 약점을 극복하는 데 도움이 될 가능성이 크다. 그럼에도 불구하고 우리나라에서 짝 프로그래밍을 실제 수업에 적용하고 그 결과를 분석하여 발표한 연구 논문을 찾기가 어렵다. 그나마 발표된 논문들은 주로 초중등 과정을 대상으로 삼았으며, 대개는 교육학적인 성과 요소들을 평가하기 위해 프로그래밍 교육 과정 자체는 비교적 단순화한 실험 목적의 수업 모델을 수립하고 실험을 수행했다. 이 연구에서는 대학에서 운영되는 실제 수업에 짝 프로그래밍을 적용한 경험과 데이터 그리고 분석, 검토 결과를 보고하였다. 이 논문이 국내의 여러 프로그래밍 입문 수업에 참고가 되기를 기대한다.

References

- [1] Alistair Cockburn and Laurie Williams(2000). The Costs and Benefits of Pair Programming. Proceedings of the First International Conference on Extreme Programming and Flexible Processes in Software Engineering (XP2000), 223-247.
- [2] Hanna Hulkko and Pekka Abrahamsson(2005). A multiple case study on the impact of pair programming on product quality. Proceeding of the 27th International Conference on Software Engineering (ICSE2005), 495-504.
- [3] Il-Jae Ko(2006). The Effect of Pair Programming on Student Achievement and Interest in Programming Class. MS dissertation, Graduate School of Education, Chung-ang University.
- [4] Jeongrang Kim(2016). Effects of Pair Programming on Creative Problem-Solving Ability and Efficiency. *Journal of The Korea Association of Information Education*, 20(1), 21-28.
- [5] Jin Kyung Kim(2010). Effects that Application of

- Pair Programming and Reflective Journal have on Self-regulated Learning and Programming Ability. MS dissertation, Graduate School of Education, Incheon National University.
- [6] Jo E. Hannay, Erik Arisholm Harald Engvik, and Dag I.K. Sjøberg(2010). Effects of Personality on Pair Programming. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, 36(1), 61-80.
- [7] Kent Beck(2000). *Extreme Programming Explained: Embrace Change*. Addison-Wesley Longman Publishing.
- [8] Keun-Woo Han, Eun-Kyoung Lee, and YoungJun Lee(2006). The Effects of Pair Programming on Achievement and Motivated Strategies in Programming Course. *Journal of The Korean Association of Computer Education*, 9(6), 19-28.
- [9] Kim Man Lui and Keith C.C. Chan(2006). Pair programming productivity: Novice - novice vs. expert - expert. *Int. J. Human-Computer Studies*, 64, 915 - 926.
- [10] Laurie Williams, D. Scott McCrickard, Lucas Layman, Khaled Hussein(2008). Eleven Guidelines for Implementing Pair Programming in the Classroom. *Agile 2008 Conference*, 445-452
- [11] Norsaremah Salleh, Emilia Mendes, and John C. Grundy(2011). Empirical Studies of Pair Programming for CS/SE Teaching in Higher Education: A Systematic Literature Review. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, 37(4), 509-525.
- [12] Radhakrishnan P. and Kanmani S.(2012). Student's Opinion on Adopting Pair Programming as a Teaching and Learning Tool. *International Journal of Computer Applications*, 60(4), 9-15.
- [13] Shamim Khan, Lydia Ray, Aurelia Smith and Angkul Kongmunvattana(2008). A Pair Programming Trial in the CS1 Lab. *ACM SIGCSE Bulletin*, 40(2), 93-96.
- [14] So Eun Jeon(2009). Evaluating the Effectiveness of Pair Programming. MS dissertation, Graduate School, Konkuk University.
- [15] Tore Dybå, Erik Arisholm, Dag I.K. Sjøberg, Jo E. Hannay, and Forrest Shull(2007). Are Two Heads Better than One? On the Effectiveness of Pair Programming. *IEEE Software*, 24(6), 12-15.
- [16] http://dmrl.kangwon.ac.kr/lecture/1602/java/1602_java.html.

저자소개

정 총 교



1995~현재 강원대학교 컴퓨터학부 교수

1989~1995 LG정보통신(주)

관심분야: 인터넷, 통신망성능분석, 시스템프로그래밍, 소프트웨어 품질, 소프트웨어 교육

e-mail: ckjeong@kangwon.ac.kr