

# 프로그래밍 초보자의 코딩 스타일을 교육하기 위한 스타일 직소

정인준\*, 이윤정\*\*, 우 균\*

\*부산대학교 컴퓨터공학과

\*\*부산대학교 U-Port 정보기술 사업단

e-mail: spd1335@gmail.com

## Style Jigsaw for Teaching the Coding Style to Novice Programmers

Jung, In-joon\*, Lee, Yun-jung\*\*, Woo Gyun\*

\*Dept of Computer Engineering, Pusan National University

\*\*Center for U-Port IT Research and Education Pusan National University

### 요 약

현대의 프로그램들은 과거의 프로그램들에 비해 복잡성이 커졌다. 그러한 변화에 따라 협력 프로젝트가 과거에 비해 일반화 되었고, 프로그램의 디버깅이 어려워졌다. 현대의 프로그래머들에게는 원활한 협력 프로젝트의 진행과 쉬운 디버깅을 위해 코딩 스타일 준수 중요성이 부각되고 있다. 프로그래머가 소스 코드를 작성할 때 코딩 스타일의 준수 여부는 가독성에 큰 영향을 미친다. 하지만 프로그래머가 코딩 스타일을 준수하지 않는다고 해도 눈에 보이는 에러가 발생하지 않는다. 그렇기 때문에 코딩 스타일은 초심자 시절의 교육이 중요하다. 이 논문에서는 효과적으로 코딩 스타일을 교육하기 위한 스타일 직소 프로그램을 구현하였다. 스타일 직소 프로그램은 코딩 스타일의 준수 정도 직소 퍼즐 형태로 보여주고, 코딩 스타일이 준수되지 않은 부분을 표시하여 줌으로써 보다 효과적으로 코딩 스타일 교육이 가능하게 한다. 본 논문에서 구현된 스타일 직소 효과를 확인하기 위해 부산대학교 컴퓨터공학과 학부 학생들을 대상으로 스타일 직소를 사용하여 코딩 스타일 교육을 실시하였다. Java 프로그래밍 교과 실습 수업에 스타일 직소 프로그램을 적용시켜 본 결과 64%의 학생이 스타일 직소가 코딩 스타일 향상에 도움이 되었다고 답하였고, 52%의 학생이 본 프로그램의 접근법이 흥미로웠다고 답하였다.

### 1. 서론

현대에 들어 컴퓨터 계산 용량과 문제의 복잡성이 급격히 증가함에 따라 프로그램들이 점점 복잡해지고 있다 [7]. 프로그램이 복잡해지면서 프로그래밍 초심자뿐만 아니라 전문가일지라도 프로그램 구조를 분석을 하기가 어려워지게 되었다. 이러한 프로그램 구조의 분석의 시간을 줄이기 위해 코딩 스타일의 중요성이 부각되고 있다.

코딩 스타일이란 소스코드가 작성되는 일반적인 형식을 의미하는데 그 범주는 매우 넓다. 예를 들어 C나 C++, Java에서 중괄호의 위치, 들여쓰기, 괄호가 사용되는 방법들 등이 코딩 스타일에 포함된다. 프로그래밍을 할 때 일정한 코딩 스타일을 사용한 소스코드와 그렇지 못한 소스코드의 가독성은 큰 차이를 보인다[1]. 이는 협동 프로젝트를 수행을 하거나 프로그램 소스코드를 디버깅을 하는 경우 효율성에 크게 영향을 미친다. 따라서 현재 코딩 스타일에 대한 다양한 연구가 진행되고 있다[2, 3, 4].

그러나 코딩 스타일을 준수하지 않는다고 해도 컴파일 에러나 런타임 에러와 같은 가시적인 에러가 발생되지는 않는다. 그렇기 때문에 코딩 스타일이 준수된 부분과 그렇지 못한 부분에 대해 명시적으로 판단하기 어렵다. 본 논

문에서는 이러한 점에 착안하여 코딩 스타일의 준수 여부를 확인하기 위해 직소 퍼즐 형태의 코딩 스타일 시각화를 제안한다. 이러한 접근법의 효과를 확인하기 위해 스타일 직소 프로그램을 구현하였으며, 실제로 대학생들의 프로그래밍 교육에 적용하여 결과를 확인하였다.

### 2. 코딩 스타일

코딩 스타일의 범주가 매우 넓은 것과 같이 코딩 스타일에 대한 표준은 다양하다. C 언어의 경우 1990년 AT&T사의 벨연구소에서 Indian Hill C Style을 발표하였고, 1997년 수정되었다[2]. 그리고 흔히 ANSI 스타일이라 일컬어지는 Eric Allman의 이름을 딴 Allman Style이 있다[5, 6]. 이 논문에서는 표준 코딩 스타일들의 규칙 중 공통적인 5가지 규칙을 골라 표 1과 같이 2가지 범주로 나누어 적용하였다.

이 논문에서는 표 1과 같이 문장과 레이아웃 두 가지 범주로 나누어 코드 스타일을 점수를 측정하였다. 문장의 범주에서는 goto문 규칙과 제어문의 중괄호 규칙을 측정하였다. 그리고 레이아웃 범주에서는 프로그램 시작 위치, 공백 문자 사용, 들여쓰기를 측정하였다.

<표 1> 본 논문에서 사용한 코딩 규칙

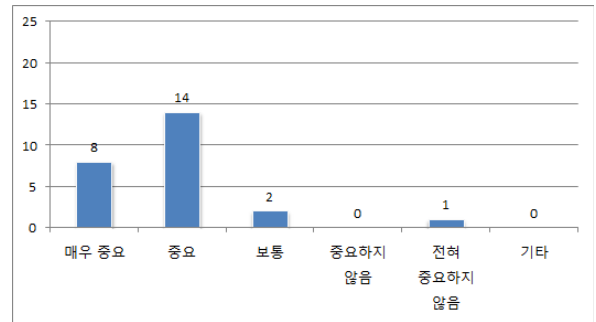
범주	코딩 규칙
문장	1. goto문을 사용하지 않는다. 2. if, else, else-if, for, while문에는 반드시 중괄호를 사용한다.
레이아웃	1. 소스코드는 문서의 첫 칸에서 시작한다. 2. if, else-if, while문에서 조건식이 들어가는 괄호 앞에는 하나의 공백문자가 들어간다. 3. 명령어 수준에 따른 일정한 들여쓰기를 사용한다.

문장 범주에서는 “goto문을 사용하지 않는다.”, “if, else, else-if, for, while문에는 반드시 중괄호를 사용한다.”라는 2가지 규칙이 포함된다. goto문은 스파게티 코드를 유발할 수 있으므로 프로그래밍에서 사용을 하지 말 것이 권장되는 명령어이다. 그리고 제어문에서 항상 중괄호를 사용할 경우 제어문의 제어 범위에 대한 가독성이 더욱 명확해진다. 레이아웃 범주에서는 “소스코드는 문서의 첫 칸에서 시작한다.”, “if, else-if, while문에서 조건식이 들어가는 괄호 앞에는 하나의 공백문자가 들어간다.”, “명령어 수준에 따른 일정한 들여쓰기를 사용한다.”라는 규칙 적용을 하였다. 소스코드의 시작 위치는 전체 소스코드의 형태에 영향을 미치며, 명령어들 사이 공백을 사용할 경우 명령어들의 구분이 명확해진다. 그리고 소스코드를 작성할 때 들여쓰기 규칙을 이용하면 소스코드의 모듈 단위를 명

확하게 나타낼 수 있으므로 소스코드의 가독성이 향상된다.

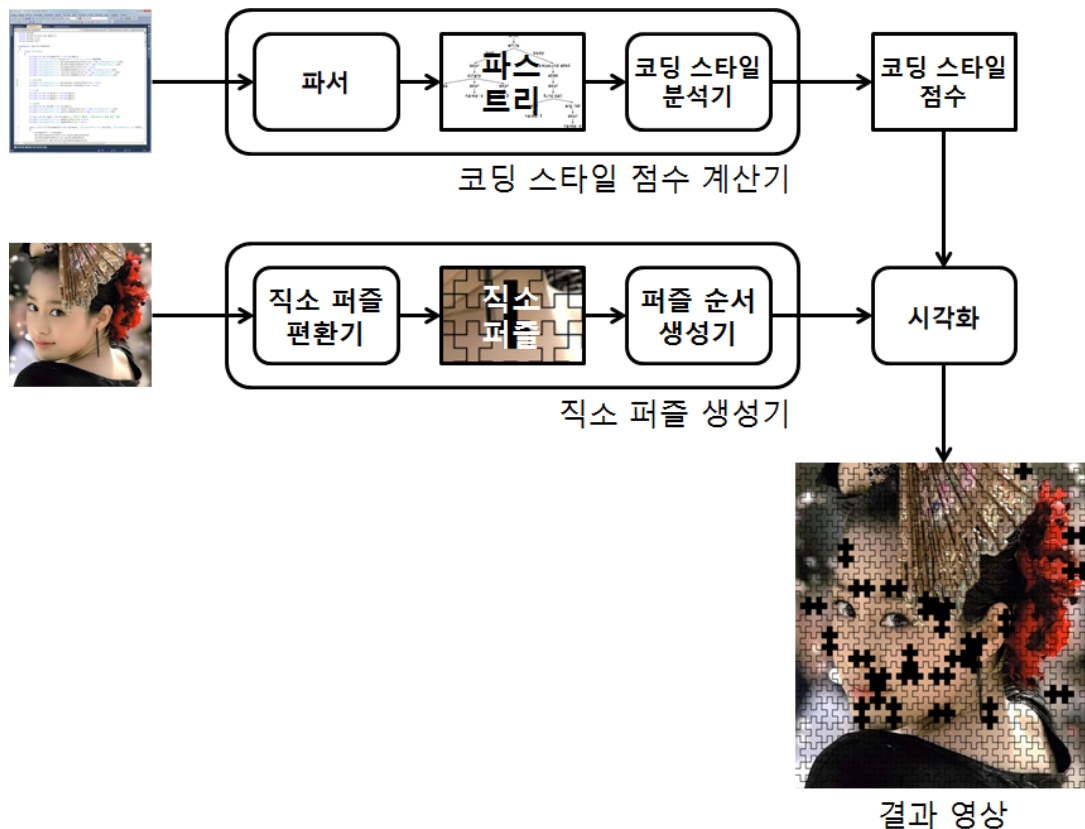
### 3. 코딩 스타일에 대한 학생들의 인식

2011년 부산대학교 Java 수업을 수강하는 컴퓨터공학과 학부생 25명을 대상으로 설문조사를 실시하였다. “코딩 스타일은 얼마나 중요하다고 생각을 하는가?”라는 질문으로 설문조사를 한 결과 그림 1과 같이 대담을 하였다.



(그림 1) “코딩 스타일은 얼마나 중요하다고 생각을 하는가?”라는 질문에 대한 학생응답 분포

25명의 학생 중 32%에 해당하는 8명의 학생이 코딩 스타일은 매우 중요하다고 답하였다. 그리고 56%에 해당하는 14명의 학생이 코딩 스타일은 중요하다고 답하였다. 반면, 중요하지 않다거나 전혀 중요하지 않다는 답변을 한



(그림 2) 전체 시스템 구성

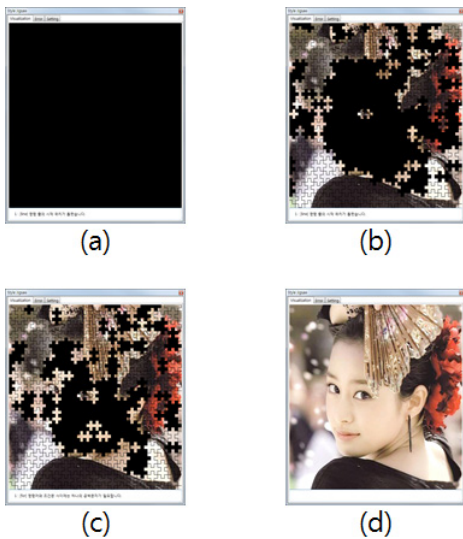
학생은 4%에 해당하는 1명의 학생뿐이었다. 즉, 학생들은 대부분 코딩 스타일의 중요성에 공감을 하고 있음을 설문조사를 통해 확인할 수 있다.

반면, 학생들의 코드를 살펴 본 결과 학생들은 소스코드를 작성할 때 코드 스타일을 잘 준수되지 않음을 확인할 수 있었다. 저자들은 이러한 원인이 코드 스타일의 경우 준수를 하지 않더라도 컴파일 에러나 런타임 에러와 같은 가시적인 에러를 보여주지 않기 때문이라고 파악하였다. 이러한 한계를 극복하기 위해 코딩 스타일 시각화라는 방법을 이용하였다.

#### 4. 스타일 직소 구현

이 논문에서는 코딩 스타일을 직소 퍼즐 형태로 시각화를 하여 코딩 스타일 교육에 도움을 주는 프로그램을 구현하였다. 스타일 직소 프로그램은 윈도우 플랫폼에서 C# 언어를 이용하여 구현하였다.

전체 시스템의 구성은 그림 2와 같다. 시스템은 코드 스타일 점수 계산 부분과 영상 처리 부분으로 나누어진다. 코드 스타일 점수 계산 부분에서는 소스코드를 입력으로 받아서 분석한 후 스타일 점수를 계산한다. 그리고 영상 처리 부분에서는 사용자가 선택한 영상을 직소 퍼즐 형태로 변환한다. 최종적으로 직소 퍼즐 형태의 이미지에 코드 스타일 점수를 적용시켜 최종 결과 화면을 보여준다.



(그림 3) 직소 퍼즐 형태로 시각화한 이미지

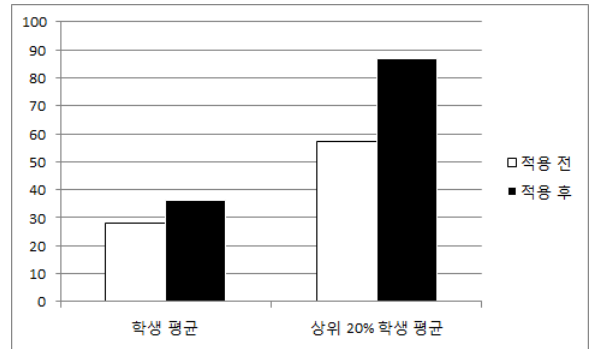
시각화 기법은 사용자가 게임과 같이 느끼면서 교육이 가능하도록 그림 3과 같은 직소 게임과 같은 형태를 고안하였다. 코드 스타일이 잘 지켜지지 않으면 그림 3의 (a)와 같이 직소 퍼즐이 적게 맞춰지게 되고, 지켜지면 (d)와 같이 직소 퍼즐이 많이 맞춰지게 된다.

그리고 직소 퍼즐의 아래에는 소스코드에서 수정해야 할 스타일 수정 사항을 하나 나타내었다. 전체 수정사항은

다른 탭을 통해 확인할 수 있다. 소스코드의 수정이 이루어지면 프로그램은 다시 소스코드를 불러와 새로 시각화 작업을 한다.

#### 5. 컴퓨터공학 전공 학생들을 대상으로 한 실험

구현된 프로그램을 부산대학교 2011년 2학기 Java 실습수업에 적용하였다. 수업의 대상은 컴퓨터공학과 학부생이었으며, 주로 2학년으로 구성되었다. 총 학생의 수는 25명이었다.



(그림 4) 스타일 직소를 적용을 하기 전/후의 학생 성적 변화

Java 실습수업은 1회 실습 당 150점 만점으로 채점되는데 스타일 직소를 사용하였을 경우와 사용하지 않았을 경우의 학생들의 평균 성적을 비교한 결과는 그림 4과 같다. 비교 지표는 전체 학생 평균과 상위 20%의 우등생 평균으로 정하였다. 스타일 직소를 사용하였을 경우 전체 평균은 약 8점이 올라갔고, 상위 20% 학생들의 평균은 약 30점 상승된 것을 확인할 수 있었다.

그리고 학생들의 코딩 스타일 교육에 도움이 되었는지 여부를 확인하기 위해 설문조사를 하였다. 설문조사는 스타일 직소를 이용한 교육이 끝난 직후에 교육을 받은 학생들을 대상으로 이루어졌다. 설문조사의 문항은 표 2와 같은 2가지이다.

<표 2> 설문조사 문항

- 스타일 직소가 코딩 스타일에 대해 이해를 하는데 도움이 되었습니까?
- 코딩 스타일에 대한 스타일 직소의 접근방법이 흥미로웠습니까?

설문조사 문항은 “스타일 직소가 코딩 스타일에 대해 이해를 하는데 도움이 되었습니까?”와 “코딩 스타일에 대한 스타일 직소의 접근방법이 흥미로웠습니까?”이다. 첫 번째 문항은 학생들의 코딩 스타일 교육에 스타일 직소가 도움이 되는지를 알아보기 위해 선정되었다. 그리고 두 번째 문항은 스타일 직소를 이용한 코딩 스타일 교육이 학

생들의 흥미를 유발할 수 있는가를 알아보기 위해 선정되었다. 두 가지 문항의 설문조사에 대한 결과는 표 3과 표 4와 같다.

<표 3> “스타일 직소가 코딩 스타일에 대해 이해를 하는데 도움이 되었습니까?”에 대한 응답

	학생 수 (명)	비율 (%)
매우 도움이 되었다.	4	16
도움이 되었다.	12	48
보통	4	16
도움이 되지 않았다.	1	4
전혀 도움이 되지 않았다.	2	8
기타	2	8
전체	25	100

“스타일 직소가 코딩 스타일에 대해 이해를 하는데 도움이 되었습니까?”라는 설문 조사 결과 매우 도움이 되었다는 학생이 4명, 도움이 되었다는 학생이 12명이었다. 반면, 도움이 되지 않았다는 대답을 한 학생이 1명 전혀 도움이 되지 않았다는 대답을 한 학생이 2명이었다. 그 외의 보통이라는 의견은 4명, 기타 의견은 2명이었다. 즉, 교육에 참가한 학생 중 약 2/3의 학생이 코딩 스타일의 이해에 도움이 되었다고 느낀 것으로 나타났다.

<표 4> “코딩 스타일에 대한 스타일 직소의 접근방법이 흥미로웠습니까?”에 대한 응답

	학생 수 (명)	비율 (%)
매우 흥미로웠다.	5	20
흥미로웠다.	8	32
보통	8	32
흥미롭지 않았다.	0	0
전혀 흥미롭지 않았다.	1	4
기타	3	12
전체	25	100

“코딩 스타일에 대한 스타일 직소의 접근방법이 흥미로웠습니까?”라는 설문 조사 결과 매우 흥미로웠다는 학생이 5명, 흥미로웠다는 학생이 8명이었다. 반면, 흥미롭지 않았다는 대답을 한 학생은 없었다. 그리고 전혀 흥미롭지 않았다는 대답을 한 학생이 1명이었다. 그 외의 보통이라는 의견은 8명이었고, 기타 의견은 3명이었다. 즉, 교육에 참가한 학생 중 과반수의 학생이 스타일 직소의 접근방법이 흥미로웠다고 느낀 것으로 나타났다.

## 6. 결론

이 논문에서는 스타일 직소라는 이름의 코딩 스타일 분석 방법을 제안하고 이에 대한 시각화 시스템을 구현하였다. 스타일 직소는 코드의 분석 및 코드 스타일 점수를 검출해 주는 코드 스타일 점수 계산 부분과 직소 퍼즐 형태의 이미지를 만들어 주는 영상 처리 부분으로 나누어진다. 스타일 직소를 이용하면 코딩 스타일을 지킬수록 직소 퍼즐이 맞춰지는 결과를 볼 수 있으므로 사용자는 더 능동적으로 코딩 스타일을 준수하게 된다.

구현된 스타일 직소를 코딩 스타일 교육을 위해서 대학교의 학부 프로그래밍 수업에 적용시켜 보았다. 그 결과 학생들의 평균 실습 점수가 향상된 것을 확인하였다. 실습 수업을 마친 후 설문 조사를 한 결과 많은 학생들이 스타일 직소가 코딩 스타일의 이해에 도움이 된다고 답하였고, 또한 스타일 직소의 코딩 스타일에 대한 접근 방식이 흥미로웠다고 답하였다.

이러한 실험 결과를 바탕으로 우리는 스타일 직소의 접근방식이 학생들의 코딩 능력 향상 및 코딩 스타일의 교육에 도움이 된다는 결론을 얻었다. 앞으로 스타일 직소를 더욱 많은 프로그래밍 언어에 적용시키고, 학생들의 프로그래밍 교육에 사용한다면 학생들의 코딩 능력 향상에 도움이 될 것이다.

## 참고문헌

- [1] T. Tenny, “Program readability: Procedures versus comments,” IEEE Transactions on Software Engineering, 14(9):1271.1279, 1988.
- [2] L.W. Cannon, R.A. Elliott, L.W. Kirchhoff, J.H. Miller, J.M. Milner, R.W. Mitze, E.P. Schan, and N.O. Whittington, “Recommended C style and coding standards,” Bell Labs, 1990.
- [3] L. Hatton, “Language subsetting in an industrial context: A comparison of MISRA C 1998 and MISRA C 2004,” Information and Software Technology, 49(5):475.482, 2007.
- [4] Richard Stallman, “GNU Coding Standards,” Free Software Foundation, 2011.
- [5] Source Forge, “Artistic Style - A Free, Fast and Small Automatic Formatter for C, C++, C#, and Java Source Code,” <http://astyle.sourceforge.net/>, 2012년 3월 13일 방문.
- [6] Open Standards, “Rationale for International Standard Programming Languages C Revision 5.10,” 2003, <http://www.open-std.org/jtc1/sc22/wg14/www/C99RationaleV5.10.pdf>, 2012년 3월 13일 방문.
- [7] F. L. Bauer, “The 1968/69 NATO Software Engineering Reports,” NATO Science Committee, 1968.