

(19) 대한민국특허청(KR)
(12) 공개특허공보(A)(11) 공개번호 10-2019-0119794
(43) 공개일자 2019년10월23일

(51) 국제특허분류(Int. Cl.)
G06F 8/41 (2018.01) G06F 8/30 (2018.01)
G09B 19/00 (2006.01) G09B 5/02 (2006.01)
(52) CPC특허분류
G06F 8/41 (2013.01)
G06F 8/30 (2013.01)
(21) 출원번호 10-2018-0043194
(22) 출원일자 2018년04월13일
심사청구일자 2018년04월13일

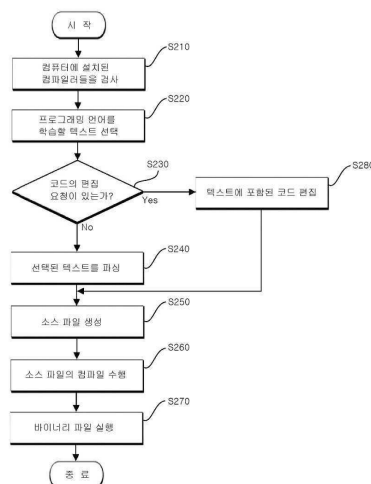
(71) 출원인
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
최진영
서울특별시 노원구 동일로248길 16, 506동 1003호 (상계동, 수락파크빌아파트)
이지성
서울특별시 은평구 갈현로25길 14-1, 402호 (갈현동, 대원아트빌)
(74) 대리인
윤귀상

전체 청구항 수 : 총 12 항

(54) 발명의 명칭 **프로그래밍 언어 학습을 위한 전자책 뷰어 및 구동 방법, 상기 방법을 수행하기 위한 저장 매체**

(57) 요약

본 발명은 프로그래밍 언어를 학습하는 이들에게 편의를 제공하는 프로그래밍 언어 학습을 위한 전자책 뷰어 및 구동 방법, 상기 방법을 수행하기 위한 저장 매체에 관한 것으로, 전자책 뷰어가 실행되면, 컴퓨터에 설치된 컴파일러들을 검사하는 단계; 실행된 전자책 뷰어에 기재된 텍스트 중에서 프로그래밍 언어를 학습할 텍스트를 선택받는 단계; 선택받은 텍스트를 파싱하여 프로그래밍 언어와 키워드를 검사하는 단계; 파싱된 텍스트의 정보를 이용하여 소스 파일을 생성하는 단계; 커맨드 라인(cmd)을 통해 생성된 소스 파일의 컴파일을 수행하는 단계; 및 컴파일의 수행이 완료되어 소스 파일과 동일한 이름의 바이너리(binary) 파일이 생성되면, 생성된 바이너리 파일을 실행시키는 단계를 포함한다.

대 표 도 - 도2

(52) CPC특허분류

G09B 19/0053 (2013.01)

G09B 5/02 (2013.01)

명세서

청구범위

청구항 1

전자책 뷰어가 실행되면, 컴퓨터에 설치된 컴파일러들을 검사하는 단계;

실행된 전자책 뷰어에 기재된 텍스트 중에서 프로그래밍 언어를 학습할 텍스트를 선택받는 단계;

선택받은 텍스트를 파싱하여 프로그래밍 언어와 키워드를 검사하는 단계;

파싱된 텍스트의 정보를 이용하여 소스 파일을 생성하는 단계;

커맨드 라인(cmd)을 통해 생성된 소스 파일의 컴파일을 수행하는 단계; 및

컴파일의 수행이 완료되어 소스 파일과 동일한 이름의 바이너리(binary) 파일이 생성되면, 생성된 바이너리 파일을 실행시키는 단계를 포함하는, 프로그래밍 언어 학습을 위한 전자책 뷰어 구동 방법.

청구항 2

제1항에 있어서, 상기 선택받은 텍스트를 파싱하여 프로그래밍 언어와 키워드를 검사하는 단계는,

컴파일러가 존재하는 경우 프로그래밍 언어와 필수 키워드의 유무를 판독하고, 컴파일러가 존재하지 않을 경우 컴파일러의 설치 여부에 대한 선택을 사용자에게 요청하는, 프로그래밍 언어 학습을 위한 전자책 뷰어 구동 방법.

청구항 3

제1항에 있어서, 상기 소스 파일을 생성하는 단계는,

파싱 완료 후, 판별된 프로그래밍 언어에 해당하는 필수 키워드가 존재하지 않는 경우, 오류 메시지를 출력하는, 프로그래밍 언어 학습을 위한 전자책 뷰어 구동 방법.

청구항 4

제1항에 있어서,

상기 텍스트를 선택받는 단계 후에, 선택받은 텍스트에 포함된 코드를 편집하는 단계를 더 포함하는, 프로그래밍 언어 학습을 위한 전자책 뷰어 구동 방법.

청구항 5

제4항에 있어서, 상기 코드를 편집하는 단계는,

선택받은 텍스트에 메인 메소드(main method)를 포함한 클래스(class)가 존재하는 경우, 선택된 텍스트를 코드를 편집하기 위한 편집 프로그램인 코드 편집기(code editor)에 출력하는, 프로그래밍 언어 학습을 위한 전자책 뷰어 구동 방법.

청구항 6

제4항에 있어서, 상기 코드를 편집하는 단계는,

선택받은 텍스트에 메인 메소드를 포함한 클래스가 없고 정적 메소드(static method)만 존재하는 경우, 임의로 클래스를 생성하면서, 생성된 클래스 안에 메인 메소드와 정적 메소드를 포함시키고, 정적 메소드를 호출하는 문을 자동으로 추가시키는, 프로그래밍 언어 학습을 위한 전자책 뷰어 구동 방법.

청구항 7

제6항에 있어서, 상기 코드를 편집하는 단계는,

임의의 클래스를 생성하여 메인 메소드와 선택된 텍스트인 정적(static) 메소드를 추가하는 단계;

추가된 정적 메소드를 분석하는 단계;

분석된 정적 메소드를 호출하는 단계;

호출된 정적 메소드의 결과형이 기본형 또는 스트링(string)형일 경우, 리턴값을 받을 지역변수를 추가하는 단계;

추가된 지역변수의 다음 줄에 해당 지역변수에 대입된 값을 출력하기 위한 명령문을 추가하는 단계; 및

정적 메소드에 매개변수가 있을 경우, '매개변수 개수 - 1'의 수만큼 콤마(,)의 추가 및 하이라이트 시키는 단계를 포함하는, 프로그래밍 언어 학습을 위한 전자책 뷰어 구동 방법.

청구항 8

제4항에 있어서, 상기 코드를 편집하는 단계는,

선택받은 텍스트에 메인 메소드를 포함한 클래스가 없고 일반 메소드만 존재하는 경우, 일반 메소드를 포함하고 있는 임의의 클래스를 생성하고, 메인 메소드를 포함하고 있는 임의의 클래스를 하나 더 생성하면서 메인 메소드 내에서 일반 메소드를 포함하고 있는 클래스에 대한 객체를 'new 연산'으로 생성시킨 후 '객체.메소드()'의 형식으로 호출하는, 프로그래밍 언어 학습을 위한 전자책 뷰어 구동 방법.

청구항 9

제8항에 있어서, 상기 코드를 편집하는 단계는,

임의의 클래스를 생성하여 선택된 텍스트인 일반 메소드를 추가하는 단계;

임의의 클래스를 하나 더 생성하여 메인 메소드를 추가하는 단계;

선택된 일반 메소드를 분석하는 단계;

추가된 메인 메소드 내에서 일반 메소드를 포함하고 있는 클래스에 대한 객체를 'new 연산'으로 생성시키는 단계;

생성된 객체를 '객체.메소드()'의 형식으로 호출하는 단계;

일반 메소드의 결과형이 기본형 또는 스트링(string)형일 경우, 리턴값을 받을 지역변수를 추가하는 단계;

추가된 지역변수의 다음 줄에 해당 지역변수에 대입된 값을 출력하기 위한 명령문을 추가하는 단계; 및

일반 메소드에 매개변수가 있을 경우, '매개변수 개수 - 1'의 수만큼 콤마(,)의 추가 및 하이라이트 시키는 단계를 포함하는, 프로그래밍 언어 학습을 위한 전자책 뷰어 구동 방법.

청구항 10

제4항에 있어서, 상기 코드를 편집하는 단계는,

선택받은 텍스트에 메인 메소드를 포함하지 않는 클래스만 존재하는 경우, 메인 메소드를 포함하고 있는 임의의

클래스를 생성하면서 메인 메소드 내에서 선택된 클래스에 대한 객체를 'new 연산'으로 생성시키는 문을 자동으로 추가하는, 프로그래밍 언어 학습을 위한 전자책 뷰어 구동 방법.

청구항 11

제1항 내지 제10항 중 어느 하나의 항에 따른 프로그래밍 언어 학습을 위한 전자책 뷰어 구동 방법을 수행하기 위한, 컴퓨터 프로그램이 기록된 컴퓨터로 판독 가능한 기록 매체.

청구항 12

전자책 뷰어가 실행되면, 컴퓨터에 설치된 컴파일러들을 검사하는 컴파일러 검사부;

전자책 뷰어에 기재된 텍스트 중에서 프로그래밍 언어를 학습할 텍스트를 선택받는 텍스트 선택부;

선택된 텍스트를 파싱(parsing)하여 프로그래밍 언어와 필수 키워드를 검사하는 텍스트 파싱부;

파싱된 텍스트의 정보를 이용하여 소스 파일을 생성하는 소스 파일 생성부;

커맨드 라인(cmd)을 통해 생성된 소스 파일의 컴파일을 수행하는 컴파일 수행부; 및

컴파일의 수행이 완료되어 소스 파일과 동일한 이름의 바이너리 파일이 생성되면, 생성된 바이너리 파일을 실행시키는 파일 실행부를 포함하는, 프로그래밍 언어 학습을 위한 전자책 뷰어.

발명의 설명

기술 분야

[0001] 본 발명은 프로그래밍 언어 학습을 위한 전자책 뷰어 및 구동 방법, 상기 방법을 수행하기 위한 저장 매체에 관한 것으로, 더욱 상세하게는 프로그래밍 언어를 학습하는 이들에게 편의를 제공할 수 있는 프로그래밍 언어 학습을 위한 전자책 뷰어 및 구동 방법, 상기 방법을 수행하기 위한 저장 매체에 관한 것이다.

배경 기술

[0003] 일반적으로 사용자는 특정 프로그램 언어에 따른 소스 코드를 작성하고 소스 코드를 컴파일하여 실행 가능한 파일을 생성한다.

[0004] 예를 들어, 프로그램 언어가 C/C++ 언어에 상응하는 경우에는 사용자는 C/C++ 언어의 문법(syntax)에 따라 소스 코드를 작성하고 C/C++ 컴파일러를 이용하여 소스 코드를 실행 가능한 파일로 변환할 수 있고, 프로그램 언어가 Java 언어에 상응하는 경우에는 사용자는 Java 언어의 문법에 따라 소스 코드를 작성하고 Java 컴파일러를 이용하여 소스 코드를 실행 가능한 파일로 변환할 수 있다.

[0005] 이를 학습하기 위해 사용자로서는, 프로그래밍 언어를 컴퓨터 상에 타이핑하는 등의 과정을 통한 실습이 동반되어야 한다. 이에 따라, 현재 대부분의 프로그래밍 학습 방법은 교재(e-book 포함)의 내용을 확인한 다음 편집 에디터에 타이핑을 한 후, 빌드 및 실행하는 과정을 거쳐 결과를 확인하는 다소 번거로운 방법을 거치고 있다.

선행기술문헌

특허문헌

[0007] (특허문헌 0001) 한국공개특허 제10-2008-0097826호

(특허문헌 0002) 한국공개특허 제10-2013-0122406호

발명의 내용

해결하려는 과제

- [0008] 본 발명의 일측면은 별도의 컴파일러 툴(예를 들어, Visual Studio 또는 eclipse 등)을 실행하지 않고도, 전자책 내에 포함된 코드의 결과를 확인하거나 편집을 용이하게 할 수 있는 프로그래밍 언어 학습을 위한 전자책 뷰어 및 구동 방법, 상기 방법을 수행하기 위한 저장 매체를 제공한다.
- [0009] 본 발명의 기술적 과제는 이상에서 언급한 기술적 과제로 제한되지 않으며, 언급되지 않은 또 다른 기술적 과제들은 아래의 기재로부터 당업자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

- [0011] 본 발명의 일 실시예에 따른 프로그래밍 언어 학습을 위한 전자책 뷰어 구동 방법은, 전자책 뷰어가 실행되면, 컴퓨터에 설치된 컴파일러들을 검사하는 단계; 실행된 전자책 뷰어에 기재된 텍스트 중에서 프로그래밍 언어를 학습할 텍스트를 선택받는 단계; 선택받은 텍스트를 파싱하여 프로그래밍 언어와 키워드를 검사하는 단계; 파싱된 텍스트의 정보를 이용하여 소스 파일을 생성하는 단계; 커맨드 라인(cmd)을 통해 생성된 소스 파일의 컴파일을 수행하는 단계; 및 컴파일의 수행이 완료되어 소스 파일과 동일한 이름의 바이너리(binary) 파일이 생성되면, 생성된 바이너리 파일을 실행시키는 단계를 포함한다.
- [0012] 일 실시예에서, 상기 선택받은 텍스트를 파싱하여 프로그래밍 언어와 키워드를 검사하는 단계는, 컴파일러가 존재하는 경우 프로그래밍 언어와 필수 키워드의 유무를 판독하고, 컴파일러가 존재하지 않을 경우 컴파일러의 설치 여부에 대한 선택을 사용자에게 요청할 수 있다.
- [0013] 일 실시예에서, 상기 소스 파일을 생성하는 단계는, 파싱 완료 후, 판별된 프로그래밍 언어에 해당하는 필수 키워드가 존재하지 않는 경우, 오류 메시지를 출력할 수 있다.
- [0014] 일 실시예에서, 상기 텍스트를 선택받는 단계 후에, 선택받은 텍스트에 포함된 코드를 편집하는 단계를 더 포함할 수 있다.
- [0015] 일 실시예에서, 상기 코드를 편집하는 단계는, 선택받은 텍스트에 메인 메소드(main method)를 포함한 클래스(class)가 존재하는 경우, 선택된 텍스트를 코드를 편집하기 위한 편집 프로그램인 코드 편집기(code editor)에 출력할 수 있다.
- [0016] 일 실시예에서, 상기 코드를 편집하는 단계는, 선택받은 텍스트에 메인 메소드를 포함한 클래스가 없고 정적 메소드(static method)만 존재하는 경우, 임의로 클래스를 생성하면서, 생성된 클래스 안에 메인 메소드와 정적 메소드를 포함시키고, 정적 메소드를 호출하는 문을 자동으로 추가시킬 수 있다.
- [0017] 일 실시예에서, 상기 코드를 편집하는 단계는, 임의의 클래스를 생성하여 메인 메소드와 선택된 텍스트인 정적(static) 메소드를 추가하는 단계; 추가된 정적 메소드를 분석하는 단계; 분석된 정적 메소드를 호출하는 단계; 호출된 정적 메소드의 결과형이 기본형 또는 스트링(string)형일 경우, 리턴값을 받을 지역변수를 추가하는 단계; 추가된 지역변수의 다음 줄에 해당 지역변수에 대입된 값을 출력하기 위한 명령문을 추가하는 단계; 및 정적 메소드에 매개변수가 있을 경우, '매개변수 개수 - 1'의 수만큼 콤마(,)의 추가 및 하이라이트 시키는 단계를 포함할 수 있다.
- [0018] 일 실시예에서, 상기 코드를 편집하는 단계는, 선택받은 텍스트에 메인 메소드를 포함한 클래스가 없고 일반 메소드만 존재하는 경우, 일반 메소드를 포함하고 있는 임의의 클래스를 생성하고, 메인 메소드를 포함하고 있는 임의의 클래스를 하나 더 생성하면서 메인 메소드 내에서 일반 메소드를 포함하고 있는 클래스에 대한 객체를 'new 연산'으로 생성시킨 후 '객체.메소드()'의 형식으로 호출할 수 있다.
- [0019] 일 실시예에서, 상기 코드를 편집하는 단계는, 임의의 클래스를 생성하여 선택된 텍스트인 일반 메소드를 추가하는 단계; 임의의 클래스를 하나 더 생성하여 메인 메소드를 추가하는 단계; 선택된 일반 메소드를 분석하는 단계; 추가된 메인 메소드 내에서 일반 메소드를 포함하고 있는 클래스에 대한 객체를 'new 연산'으로 생성시

키는 단계; 생성된 객체를 '객체.메소드()'의 형식으로 호출하는 단계; 일반 메소드의 결과형이 기본형 또는 스트링(string)형일 경우, 리턴값을 받을 지역변수를 추가하는 단계; 추가된 지역변수의 다음 줄에 해당 지역변수에 대입된 값을 출력하기 위한 명령문을 추가하는 단계; 및 일반 메소드에 매개변수가 있을 경우, '매개변수 개수 - 1'의 수만큼 콤마(,)의 추가 및 하이라이트 시키는 단계를 포함할 수 있다.

[0020] 일 실시예에서, 상기 코드를 편집하는 단계는, 선택받은 텍스트에 메인 메소드를 포함하지 않는 클래스만 존재하는 경우, 메인 메소드를 포함하고 있는 임의의 클래스를 생성하면서 메인 메소드 내에서 선택된 클래스에 대한 객체를 'new 연산'으로 생성시키는 문을 자동으로 추가할 수 있다.

[0021] 본 발명의 다른 실시예에 따른 컴퓨터로 판독 가능한 저장 매체에는, 프로그래밍 언어 학습을 위한 전자책 뷰어 구동 방법을 수행하기 위한 컴퓨터 프로그램이 기록되어 있다.

[0022] 본 발명의 일 실시예에 따른 프로그래밍 언어 학습을 위한 전자책 뷰어는, 전자책 뷰어가 실행되면, 컴퓨터에 설치된 컴파일러들을 검사하는 컴파일러 검사부; 전자책 뷰어에 기재된 텍스트 중에서 프로그래밍 언어를 학습할 텍스트를 선택받는 텍스트 선택부; 선택된 텍스트를 파싱(parsing)하여 프로그래밍 언어와 필수 키워드를 검사하는 텍스트 파싱부; 파싱된 텍스트의 정보를 이용하여 소스 파일을 생성하는 소스 파일 생성부; 커맨드 라인(cmd)을 통해 생성된 소스 파일의 컴파일을 수행하는 컴파일 수행부; 및 컴파일의 수행이 완료되어 소스 파일과 동일한 이름의 바이너리 파일이 생성되면, 생성된 바이너리 파일을 실행시키는 파일 실행부를 포함한다.

발명의 효과

[0024] 상술한 본 발명의 일 측면에 따르면, 전자책에 실린 프로그램 코드를 실행 파일로 만들어 결과를 바로 출력하여 사용자가 즉각적으로 결과를 확인을 할 수 있도록 하며, 프로그램 코드가 전체가 아닌 일부분만 제공되었다 하더라도 상황에 맞게 코드를 재생성하여 사용자가 편리하게 학습할 수 있도록 하는 효과를 제공한다.

도면의 간단한 설명

[0026] 도 1은 본 발명의 일 실시예에 따른 프로그래밍 언어 학습을 위한 전자책 뷰어의 개략적인 구성이 도시된 도면이다.

도 2는 본 발명의 일 실시예에 따른 프로그래밍 언어 학습을 위한 전자책 뷰어 구동 방법을 설명하는 순서도이다.

도 3 내지 도 16은 컴퓨터 상에서 구현되고 있는 본 발명에 따른 실시예를 설명하는 도면이다.

도 17은 도 2에 있는 코드를 편집하는 단계를 설명하는 순서도이다.

발명을 실시하기 위한 구체적인 내용

[0027] 후술하는 본 발명에 대한 상세한 설명은, 본 발명이 실시될 수 있는 특정 실시예를 예시로서 도시하는 첨부 도면을 참조한다. 이들 실시예는 당업자가 본 발명을 실시할 수 있기에 충분하도록 상세히 설명된다. 본 발명의 다양한 실시예는 서로 다르지만 상호 배타적일 필요는 없음이 이해되어야 한다. 예를 들어, 여기에 기재되어 있는 특정 형상, 구조 및 특성은 일 실시예와 관련하여 본 발명의 정신 및 범위를 벗어나지 않으면서 다른 실시예로 구현될 수 있다. 또한, 각각의 개시된 실시예 내의 개별 구성요소의 위치 또는 배치는 본 발명의 정신 및 범위를 벗어나지 않으면서 변경될 수 있음이 이해되어야 한다. 따라서, 후술하는 상세한 설명은 한정적인 의미로서 취하려는 것이 아니며, 본 발명의 범위는, 적절하게 설명된다면, 그 청구항들이 주장하는 것과 균등한 모든 범위와 더불어 첨부된 청구항에 의해서만 한정된다. 도면에서 유사한 참조부호는 여러 측면에 걸쳐서 동일하거나 유사한 기능을 지칭한다.

[0028] 이하, 도면들을 참조하여 본 발명의 바람직한 실시예들을 보다 상세하게 설명하기로 한다.

[0030] 도 1은 본 발명의 일 실시예에 따른 프로그래밍 언어 학습을 위한 전자책 뷰어의 개략적인 구성이 도시된 도면이다.

- [0031] 도 1을 참조하면, 프로그래밍 언어 학습을 위한 전자책 뷰어는, 컴파일러 검사부(100), 텍스트 선택부(200), 텍스트 파싱부(300), 소스 파일 생성부(400), 컴파일 수행부(500) 및 파일 실행부(600)를 포함한다.
- [0032] 컴파일러 검사부(100)는, 전자책 뷰어가 실행되면, 컴퓨터에 설치된 컴파일러들을 검사한다.
- [0033] 텍스트 선택부(200)는, 전자책 뷰어에 기재된 텍스트 중에서 프로그래밍 언어를 학습할 텍스트를 선택받는다.
- [0034] 사용자가 전자책 뷰어 상의 텍스트 중에서 학습할 테스트를 입력 수단(예를 들어, 마우스 또는 키보드 등)을 통해 선택한 경우, 선택된 텍스트를 입력받아 클립 보드(clipboard, 컴퓨터에서 임시 저장 공간으로 사용하기 위해 확보된 메모리 영역)에 임시로 저장해 둘 수 있다.
- [0035] 텍스트 파싱부(300)는, 클립보드로부터 텍스트 선택부(200)에서 선택받은 데이터를 가져와서 토큰(token) 단위로 분석을 한다. 이 때 선택된 텍스트가 어떤 프로그래밍 언어인지를 알 수 있고, 최소한의 키워드들이 있는지를 검사하게 된다.
- [0036] 소스 파일 생성부(400)는, 텍스트 파싱부(300)에서 파싱된 텍스트의 정보를 이용하여 소스 파일을 생성한다.
- [0037] 일 실시예에서, 소스 파일 생성부(400)는, 변환할 프로그래밍 언어에 대한 소스 파일을 생성하기 위한 최소한의 키워드가 존재하는 경우에 한하여, 선택된 텍스트의 내용으로 소스 파일을 생성할 수 있다.
- [0038] 만약, 코드가 아닌 일반 텍스트이거나 해당 프로그래밍 언어가 아니라면 도 5에 도시된 바와 같은 경고 문구(예를 들어, "잘못된 코드입니다." 등)가 기재된 메시지 박스를 출력하여 통지할 수 있다.
- [0039] 컴파일 수행부(500)는, 커맨드 라인(cmd)을 통해 소스 파일 생성부(400)에서 생성된 소스 파일의 컴파일(Compile, 고급언어(CHILL, COBOL, FORTRAN 등으로 인간이 구분하기 쉬운 언어)로 작성된 프로그램을 기계어(컴퓨터 등의 기계가 이해할 수 있는 언어)로 번역하는 것)을 수행한다.
- [0040] 컴파일의 수행이 완료되면, 소스 파일과 동일한 이름의 바이너리 파일(예를 들어, C 또는 C#이라면 exe 파일이 생성되고, Java인 경우에는 class 파일이 생성)이 생성된다.
- [0041] 파일 실행부(600)는, 컴파일 수행부(500)에서 컴파일의 수행이 완료되어 소스 파일과 동일한 이름의 바이너리 파일이 생성되면, 생성된 바이너리 파일을 실행시킨다.
- [0042] 일 실시예에서, 파일 실행부(600)는, 컴파일 과정이 끝날 때까지 대기를 한 후에(예를 들어, WaitForSingleObject 등과 같은 함수를 사용), 바이너리 파일이 정상적으로 생성 되었다면 해당 파일을 실행시키고, 그렇지 않다면 오류 메시지를 출력할 수 있다.
- [0043] 상술한 바와 같은 구성을 가지는 프로그래밍 언어 학습을 위한 전자책 뷰어는, 운영체제(Operation System: OS), 즉 시스템을 기반으로 다양한 소프트웨어를 실행하거나 제작할 수 있다. 상기 운영체제는 소프트웨어가 장치의 하드웨어를 사용할 수 있도록 하기 위한 시스템 프로그램으로서, 안드로이드 OS, iOS, 윈도우 모바일 OS, 바다 OS, 심비안 OS, 블랙베리 OS 등 모바일 컴퓨터 운영체제 및 윈도우 계열, 리눅스 계열, 유닉스 계열, MAC, AIX, HP-UX 등 컴퓨터 운영체제를 모두 포함할 수 있다.
- [0044] 상술한 바와 같은 구성을 가지는 프로그래밍 언어 학습을 위한 전자책 뷰어의 각 구성에 의한 구동 방법은 도 2 이하의 방법 설명에서 후술한다.
- [0046] 도 2는 본 발명의 일 실시예에 따른 프로그래밍 언어 학습을 위한 전자책 뷰어 구동 방법을 설명하는 순서도이다.
- [0047] 도 2를 참조하면, 프로그래밍 언어 학습을 위한 전자책 뷰어 구동 방법은, 우선 전자책 뷰어가 실행되면, 컴퓨터에 설치된 컴파일러들을 검사한다(S210).
- [0048] 사용자가 전자책 뷰어를 실행한 후, 전자책의 지문 중에 프로그래밍 언어를 학습할 텍스트가 기재되어 있는 부분을 마우스나 키보드 등의 입력 수단을 이용하여 프로그래밍 언어를 학습할 텍스트를 선택한다(S220).
- [0049] 상술한 단계 S220에서 텍스트를 선택받은 후, 사용자로부터 선택된 텍스트에 포함된 코드의 편집에 대한 요청이 없는 경우(S230의 No의 경우), 클립보드를 통해 구한 텍스트 정보를 토큰(token) 단위로 파싱(parsing)하여 어떤 프로그래밍 언어인지, 해당 언어에 필요한 최소한의 키워드는 있는지를 분석한다(S240)
- [0050] 상술한 단계 S220에서 텍스트를 선택한 후 도 3에 도시된 바와 같이 메뉴나 톨바로 "프로그램 실행"을

선택하면, 파싱 과정에서 분석한 프로그래밍 언어에 해당하는 컴파일러(compiler)가 존재할 경우 해당 텍스트를 소스 파일로 저장한(S250) 뒤 빌드(S260)하여 바로 프로그램을 실행하게 된다(S270).

- [0051] 즉, 상술한 단계 S250에서는 파싱된 텍스트의 정보를 이용하여 소스 파일을 생성하고, 상술한 단계 S260에서는 커맨드 라인(cmd)을 통해 생성된 소스 파일의 컴파일을 수행하고, 상술한 단계 S270에서는 컴파일의 수행이 완료되어 소스 파일과 동일한 이름의 바이너리(binary) 파일이 생성되면, 생성된 바이너리 파일을 실행하게 된다.
- [0052] 상술한 단계 S240에서, 프로그래밍 언어에 해당하는 컴파일러가 존재하는 경우 프로그래밍 언어와 필수 키워드의 유무를 판독하고, 컴파일러가 존재하지 않을 경우 컴파일러의 설치 여부에 대한 선택을 사용자에게 요청할 수 있다.
- [0053] 만약 컴파일러가 존재하지 않는다면, 도 4에 도시된 바와 같이 해당 언어에 대한 컴파일러를 설치할 것인지를 묻는 메시지 박스를 출력되고, "예" 버튼을 눌렀을 경우는 컴파일러 설치 페이지로 이동시켜 준다. 만약, 사용자가 "아니요" 버튼을 누르게 되면, 본 발명의 구동을 종료하게 된다.
- [0054] 상술한 단계 S240에서 파싱하여 얻은 정보를 이용하여 소스 파일(source file, 원시 프로그램 코드가 포함되어 있는 파일)을 생성한다(S250).
- [0055] 상술한 단계 S250은, 파싱하여 얻은 정보로부터 프로그래밍 언어의 종류와 해당 언어를 작성하기 위한 필수 키워드의 유무를 판독한 후, 필수 키워드가 존재하는 경우, 프로그래밍 정보를 이용하여 소스 파일을 생성할 수 있다.
- [0056] 즉, 해당 프로그래밍 언어에 대한 소스 파일을 생성하기 위한 최소한의 키워드들이 있는지 체크를 하게 되는데, 예를 들어, Java는 최소 하나 이상의 클래스(class)로 이루어지며, 모든 프로그램은 반드시 메인 메소드(main method)를 가지고 있어야 한다.
- [0057] 상술한 단계 S250은, 파싱하여 얻은 정보로부터 프로그래밍 언어에 대한 소스 파일을 생성하기 위한 필수 키워드의 유무를 판독한 후, 오류 메시지를 출력할 수 있다.
- [0058] 만약, 상술한 단계 S250에서 코드가 아닌 일반 텍스트이거나 해당 프로그래밍 언어가 아니라면 도 5에 도시된 바와 같은 메시지 박스가 출력된다.
- [0059] 상술한 바와 같은 필수 키워드들에 대한 체크가 끝났다면, 선택된 텍스트의 내용으로 도 6에 도시된 바와 같은 소스 파일을 생성하게 된다.
- [0060] 본 발명에서는, Temp 폴더에 'TestProgram'이라는 하위 폴더를 생성된 후, 'main 메소드'를 가지고 있는 클래스 뒤에 오는 문자열의 이름으로 소스 파일을 저장한다.
- [0061] 상술한 단계 S250에서 소스 파일이 생성되면, 커맨드 라인(cmd)을 통해 상술한 단계 S250에서 생성된 소스 파일의 컴파일을 수행한다(S260).
- [0062] 상술한 단계 S260에서 생성된 소스 파일의 컴파일이 성공적으로 완료되면, 소스 파일과 동일한 이름의 바이너리 파일이 생성된다. 예를 들어 C 또는 C#이라면 'exe' 파일이 생성되고, Java인 경우에는 'class' 파일이 생성된다.
- [0063] 상술한 단계 S260에서 컴파일의 수행이 완료되면, 도 7에 도시된 바와 같은 소스 파일과 동일한 이름의 바이너리 파일(binary file, 인간이 판독할 수 있는 텍스트로 구성된 문서 파일과는 달리, 이진수 데이터 또는 실행 명령어들로 구성된 파일)이 정상적으로 생성되면, 생성된 바이너리 파일을 실행시킨다(S270).
- [0064] 컴파일 과정이 끝날 때까지 대기를 한 후(WaitForSingleObject 같은 함수를 사용), 바이너리 파일이 정상적으로 생성 되었다면 도 8에 도시된 바와 같이 해당 파일이 실행되어(Java의 경우라면 Virtual Machine 위에 실행) 출력 결과를 띄어 주고, 그렇지 않다면 도 9에 도시된 바와 같이 "error" 등의 오류 메시지를 출력하게 된다.
- [0065] 상술한 바와 같은 단계를 가지는 프로그래밍 언어 학습을 위한 전자책 뷰어 구동 방법은, 상술한 텍스트를 선택 받는 단계(S220) 후에, 사용자로부터 선택된 텍스트에 포함된 코드의 편집에 대한 요청이 있는 경우(S230의 Yes의 경우), 선택받은 텍스트에 포함된 코드를 편집하는 단계(S280)를 더 포함할 수 있다.
- [0066] 도 10에 도시된 바와 같이, 전자책 뷰어에서 코드가 있는 부분의 텍스트를 선택한 후, 메뉴나 툴바 등을 이용하여 "코드 편집 및 실행"을 선택하게 되면, 코드를 편집하는 단계(S280)를 수행하게 된다.
- [0067] 이하에서는 설명의 편의상 Java의 경우를 예로 들어 설명한다. 다만, C#도 Java와 거의 동일하게 구현이 되며,

C언어의 경우에는 main 함수만 체크하면 되므로 더 간단하게 구현될 수 있을 것이다.

- [0068] 먼저, 코드를 편집하는 단계(S280)는, 도 11에 도시된 바와 같이 선택받은 텍스트에 메인 메소드(main method)를 포함한 클래스(class)가 존재하는 경우, 선택된 텍스트를 코드를 편집하기 위한 편집 프로그램인 코드 편집기(code editor)에 출력할 수 있다.
- [0069] 선택한 텍스트에 메인 메소드를 포함한 클래스가 있는 소스 파일은, 그 자체만으로도 온전하게 컴파일이 될 수 있다. 이 경우에는 선택된 텍스트가 그대로 코드 편집기에 출력된다.
- [0070] 사용자가 이에 따라 "실행" 버튼을 누르면 상술한 S250 내지 S270의 과정을 거쳐 결과를 확인할 수 있고, 또는, 일부 코드를 사용자가 원하는 대로 편집해서 실행할 수도 있다.
- [0071] 다음으로, 코드를 편집하는 단계(S280)는, 도 12에 도시된 바와 같이 선택받은 텍스트에 메인 메소드를 포함한 클래스가 없고 정적 메소드(static method)만 존재하는 경우, 메인 메소드를 포함하고 있는 클래스가 없기 때문에, 그 자체만으로 컴파일 되어 실행될 수가 없다. 따라서 도 13에 도시된 바와 같이 임의로 클래스를 생성하면서 생성된 클래스 안에 메인 메소드와 정적 메소드를 포함시키고, 정적 메소드를 호출하는 문을 자동으로 추가시킨다.
- [0072] 일 실시예에서, 코드를 편집하는 단계(S280)는, 선택받은 텍스트에 메인 메소드를 포함한 클래스가 없고 정적 메소드(static method)만 존재하는 경우, 먼저 임의의 클래스를 생성하여 메인 메소드와 선택된 텍스트인 정적(static) 메소드를 추가하고, 추가된 정적 메소드를 분석하고, 분석된 정적 메소드를 호출하고, 호출된 정적 메소드의 결과형이 기본형 또는 스트링(string)형일 경우, 리턴값을 받을 지역변수를 추가하고, 추가된 지역변수의 다음 줄에 해당 지역변수에 대입된 값을 출력하기 위한 명령문을 추가한 후, 정적 메소드에 매개변수가 있을 경우, '매개변수 개수 - 1'의 수만큼 콤마(,)의 추가 및 하이라이트 시킬 수 있다.
- [0073] 즉, 도 13에 도시된 바와 같이, SampleProgram이라는 클래스를 생성하면서 메인 메소드를 추가하며, 도 12에서 선택한 텍스트인 minFunction 메소드를 분석하여, 정적 메소드의 결과형이 기본형(예를 들어, byte, short, int, long, char, float, double 또는 boolean 등) 또는 스트링형일 경우에는 정적 메소드를 호출한 뒤 리턴값을 받을 지역변수를 추가하고(int val = minFunction();), 그 다음 줄에서 그 변수에 대입된 값을 출력하기 위한 간단한 명령문(System.out.println(val);)이 추가된다.
- [0074] 만약, 결과형이 void나 그 외의 자료형일 경우에는 단순히 정적 메소드를 호출하는 부분만 추가된다.
- [0075] 일 실시예에서, 코드를 편집하는 단계(S280)는, 메소드의 매개변수의 개수를 파악한 후, 코드 편집기가 출력될 때 메소드 안에 '매개변수의 개수 - 1'의 개수만큼의 콤마(,)를 출력하고, 출력된 콤마를 하이라이트 시켜줄 수 있다.
- [0076] 이때, 메소드에 매개변수가 존재하는 경우, 해당 메소드를 호출하는 부분 뒤에 "Please input parameter"라는 주석을 달아주며, 매개변수 입력 부분에 자동으로 포커스가 이동되어 있기 때문에 사용자는 바로 입력하여 편하게 결과를 확인할 수 있다.
- [0077] 도 14는 매개변수 값으로 '10'과 '5'를 입력한 후, "실행" 버튼을 누른 경우이다.
- [0078] 다음으로, 코드를 편집하는 단계(S280)는, 도 15에 도시된 바와 같이 선택받은 텍스트에 메인 메소드를 포함한 클래스가 없고 일반 메소드만 존재하는 경우, 메인 메소드를 포함하고 있는 클래스가 없기 때문에, 그 자체만으로 컴파일 되어 실행할 수 없다. 이 경우 가능한 방법이 2가지가 있다.
- [0079] 첫 번째 방법은, 임의로 클래스를 생성하면서 그 안에 메인 메소드와 일반 메소드를 포함시킨 후, 메인 메소드 내에서 추가된 클래스에 해당하는 객체를 'new 연산'을 통해 생성한 뒤 '객체.메소드()'의 형식으로 호출하는 문을 자동으로 추가시키는 것이다.
- [0080] 두 번째 방법은, 일반 메소드를 포함하고 있는 임의의 클래스를 생성한 후, 메인 메소드를 포함하고 있는 임의의 클래스를 하나 더 생성하면서 메인 메소드 내에서 일반 메소드를 포함하고 있는 클래스에 대한 객체를 'new 연산'으로 생성시키고 '객체.메소드()'의 형식으로 호출하는 것이다.
- [0081] 본 발명에서는, 두 방법 중 보다 일반적인 두 번째 방법을 사용했다.
- [0082] 일 실시예에서, 코드를 편집하는 단계(S280)는, 선택받은 텍스트에 메인 메소드를 포함한 클래스가 없고 일반 메소드만 존재하는 경우, 임의의 클래스를 생성하여 선택된 텍스트인 일반 메소드를 추가하고, 임의의 클래스를

하나 더 생성하여 메인 메소드를 추가하고, 선택된 일반 메소드를 분석하고, 추가된 메인 메소드 내에서 일반 메소드를 포함하고 있는 클래스에 대한 객체를 'new 연산' 으로 생성시키고, 생성된 객체를 '객체.메소드()' 의 형식으로 호출하고, 일반 메소드의 결과형이 기본형 또는 스트링(string)형일 경우, 리턴값을 받을 지역변수를 추가하고, 추가된 지역변수의 다음 줄에 해당 지역변수에 대입된 값을 출력하기 위한 명령문을 추가한 후, 일반 메소드에 매개변수가 있을 경우, '매개변수 개수 - 1'의 수만큼 콤마(,)의 추가 및 하이라이트 시킬 수 있다.

- [0083] 이 이후의 과정은 상술한 선택받은 텍스트에 메인 메소드를 포함한 클래스가 없고 정적 메소드(static method)만 존재하는 경우와 동일하므로 생략한다.
- [0084] 마지막으로, 코드를 편집하는 단계(S280)는, 도 16에 도시된 바와 같이 메인 메소드를 포함하지 않는 클래스만 존재하는 경우이다.
- [0085] 선택한 텍스트가 메인 메소드를 포함하고 있는 클래스가 없기 때문에, 그 자체만으로 컴파일 되어 실행할 수 없다. 따라서 메인 메소드를 포함하고 있는 임의의 클래스를 생성하면서 메인 메소드 내에서 선택된 텍스트의 클래스에 대한 객체를 'new 연산'을 통해 생성시키는 문을 추가시킨다.
- [0086] 다만, 보통 클래스가 정의된 경우에는 메소드가 여러 개인 경우가 대부분이므로 특별히 메소드를 호출하는 문은 생성하지 않고, 메소드 호출 부분은 사용자가 직접 입력하도록 한다.
- [0087] 도 17을 참조하면, 코드를 편집하는 단계(S280)의 전체적인 과정은, 먼저 사용자로부터 코드의 편집 요청이 있는 경우(S220의 Yes의 경우), 메인 메소드를 포함한 클래스가 있는 경우(S281의 Yes의 경우) 곧바로 코드 편집기를 출력(S285)하게 되고, 메인 메소드를 포함한 클래스가 없는 경우(S281의 No의 경우) 클래스의 유무를 판단(S282)하게 된다.
- [0088] 클래스가 있는 경우(S282의 Yes의 경우) 메인 메소드를 포함한 클래스 생성 및 'new 연산'을 통해 객체를 생성(S286)하게 되고, 클래스가 없는 경우(S282의 No의 경우) 정적 메소드의 유무를 판단(S283)하게 된다.
- [0089] 정적 메소드가 있는 경우(S283의 Yes의 경우) 메인 메소드와 정적 메소드를 포함한 클래스를 생성하고 정적 메소드를 호출(S287)하고, 정적 메소드가 없는 경우(S283의 No의 경우) 일반 메소드의 유무를 판단(S284)하게 된다.
- [0090] 일반 메소드가 있는 경우(S284의 Yes의 경우) 메인 메소드를 포함한 클래스의 생성 및 'new 연산'을 통한 객체 생성과 '객체.메소드()'를 호출(S288)하고, 일반 메소드마저 없는 경우(S284의 Yes의 경우) 잘못된 코드로 판단(S289)하여 구동을 종료하게 된다.
- [0091] 상술한 바와 같은 단계를 가지는 프로그래밍 언어 학습을 위한 전자책 뷰어 구동 방법은, 전자책 뷰어의 종료시, 상술한 단계를 거치면서 임시파일로 저장되었거나 임시적으로 생성한 파일을 삭제하는 단계(설명의 편의상 도면에는 도시하지 않음)를 더 포함할 수 있다.
- [0092] 프로그래밍 언어를 학습하기 위해서는 실습이 동반되어야 한다. 그러나 현재 대부분의 프로그래밍 학습 방법은 교재(e-book 포함)의 내용을 확인한 다음 편집 에디터에 타이핑을 한 후, 빌드 및 실행하는 과정을 거쳐 결과를 확인하는 다소 번거로운 방법을 거치고 있다.
- [0093] 이에 따라, 본 발명에서는, 기존의 타이핑하는 과정을 줄이고, 바로 실행을 가능하게 함으로써, 편하고 신속하게 결과를 확인할 수 있다.
- [0094] 뿐만 아니라, 지문에 전체 코드가 아닌 일부분만 실린 경우에도 상황에 맞는 코드를 생성해 줌으로써, 역시 빠른 결과의 확인이 가능하며 초보자의 경우에는 코드 작성에 도움을 줄 수도 있다.
- [0096] 상술한 바와 같은 프로그래밍 언어 학습을 위한 전자책 뷰어 구동 방법은, 응용 프로그램(application)으로 구현되거나 다양한 컴퓨터 구성요소를 통하여 수행될 수 있는 프로그램 명령어의 형태로 구현되어 컴퓨터 판독 가능한 기록 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능한 기록 매체는 프로그램 명령어, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다.
- [0097] 상기 컴퓨터 판독 가능한 기록 매체에 기록되는 프로그램 명령어는 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 분야의 당업자에게 공지되어 사용 가능한 것일 수도 있다.

- [0098] 컴퓨터 관독 가능한 기록 매체의 예에는, 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체, CDROM, DVD 와 같은 광기록 매체, 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 ROM, RAM, 플래시 메모리 등과 같은 프로그램 명령어를 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다.
- [0099] 프로그램 명령어의 예에는, 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드도 포함된다. 상기 하드웨어 장치는 본 발명에 따른 처리를 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.
- [0101] 전자책(e-book)이란 "도서로 간행되었거나 간행될 수 있는 저작물의 내용이 디지털 데이터를 이용해 전자 기록 매체·저장 장치에 수록된 뒤, 유무선 정보통신망을 통해 컴퓨터나 휴대단말기로 그 내용을 읽고 보고 들을 수 있도록 한 디지털 도서를 총칭"한다.
- [0102] 전세계적으로 서책(종이책)의 성장률은 점차 줄어들고 있는데 반해, 전자책은 위에서 언급한 장점들로 인해 증가하고 있는 추세이다.
- [0103] 전자책 중 가장 쉽게 접할 수 있는 대표적인 형식은 pdf이며, 전자책의 한 유형인 디지털교과서는 ePub으로도 많이 제작되고 있는 편이다.
- [0104] 2007년부터 정부 주도로 추진된 디지털 교과서(Digital Textbook)는 "서책형 교과서에 멀티미디어 학습자료와 학습지원 기능 등을 부가하여 제공되는 학생용 교재를 말한다."라고 정의하고 있다.
- [0105] 디지털 교과서의 주요 기능은 기본적인 전자책의 기능 외에 네트워크를 통한 상호작용이 강화되었다.
- [0106] 디지털 교과서는 대체적으로 ePub 기반으로 개발이 되는 편이나, 다른 형식으로 개발이 되는 경우도 있다
- [0107] 정부의 소프트웨어 교육에 대한 관심은 지속적으로 늘어나고 있으며, 2018년부터는 소프트웨어 교육(코딩 교육)이 의무화된다.
- [0108] 이러한 상황이라면 추후 디지털 교과서에 프로그래밍 언어 과목이 추가될 가능성도 있어 보인다. 그러나, 프로그래밍 언어를 학습하기 위해서는 단순히 책의 내용을 보는 데 그치는 것이 아니라 "컴퓨터가 있어야 하고, 내가 배우고자 하는 언어를 편집할 수 있는 에디터 및 이를 빌드해 줄 수 있는 컴파일러가 제공되어야 한다." 이 관점에서 볼 때 기존에 전자책 뷰어(디지털 교과서 포함)가 제공하던 기능들 만으로는 프로그래밍 언어를 공부하는 데는 부족한 면이 있다. 별도로 컴파일러 툴을 실행해서 편집을 해야 하는 불편함이 있기 때문이다.
- [0109] 이에 따라, 본 발명에서는 전자책의 내용을 보는 학습자가 별도로 편집하고 빌드하여 프로그램을 실행하는 과정 없이, 뷰어에서 바로 실행할 수 있도록 하여 편의성을 제공할 수 있다.
- [0111] 이상에서는 실시예들을 참조하여 설명하였지만, 해당 기술 분야의 숙련된 당업자는 하기의 특허 청구의 범위에 기재된 본 발명의 사상 및 영역으로부터 벗어나지 않는 범위 내에서 본 발명을 다양하게 수정 및 변경시킬 수 있음을 이해할 수 있을 것이다.

산업상 이용가능성

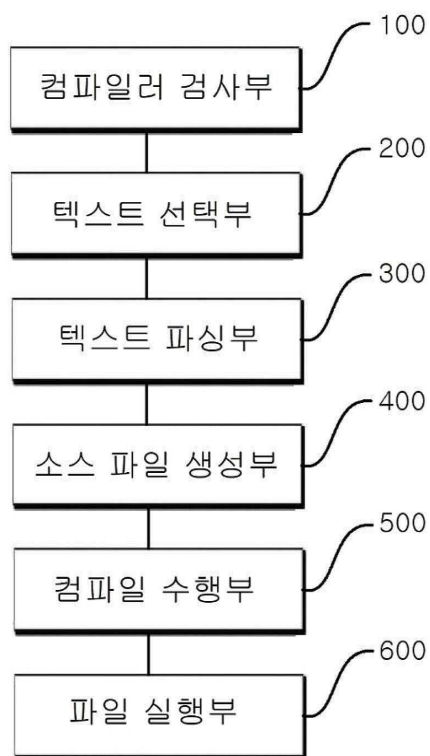
- [0113] 본 발명에 따르면, 전자책 내에 포함된 코드를 별도의 컴파일러 툴(Visual Studio, eclipse 등)을 실행하지 않고, 바로 결과를 확인할 수 있도록 편의를 제공할 수 있다.
- [0114] 또한, 전자책 내에 포함된 전체 또는 일부분의 코드를 편집하여 손쉽게 결과를 확인할 수 있도록, 코드를 자동으로 생성할 수 있다.
- [0115] 이에 따라, 전자책 뷰어나 디지털교과서 등의 개발 회사에서 프로그래밍 관련 과목에 대한 편의 기능으로 본 발명을 적용할 수 있을 것으로 예상된다.

부호의 설명

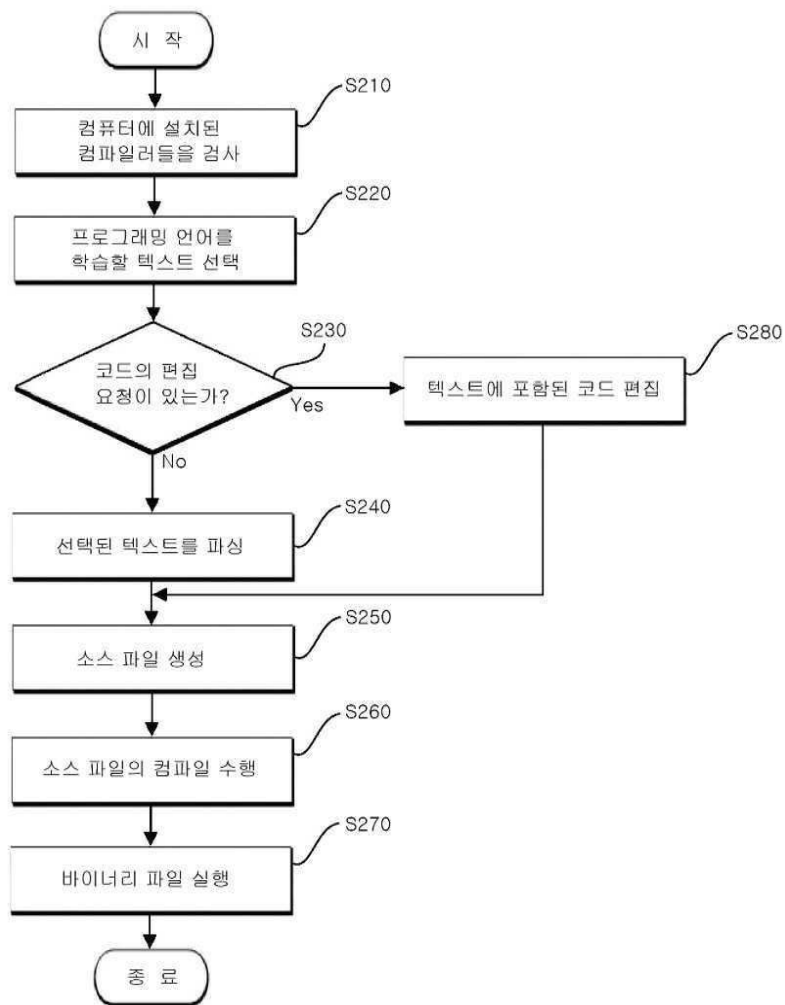
- [0117]
- 100: 컴파일러 검사부
 - 200: 텍스트 선택부
 - 300: 텍스트 파싱부
 - 400: 소스 파일 생성부
 - 500: 컴파일 수행부
 - 600: 파일 실행부

도면

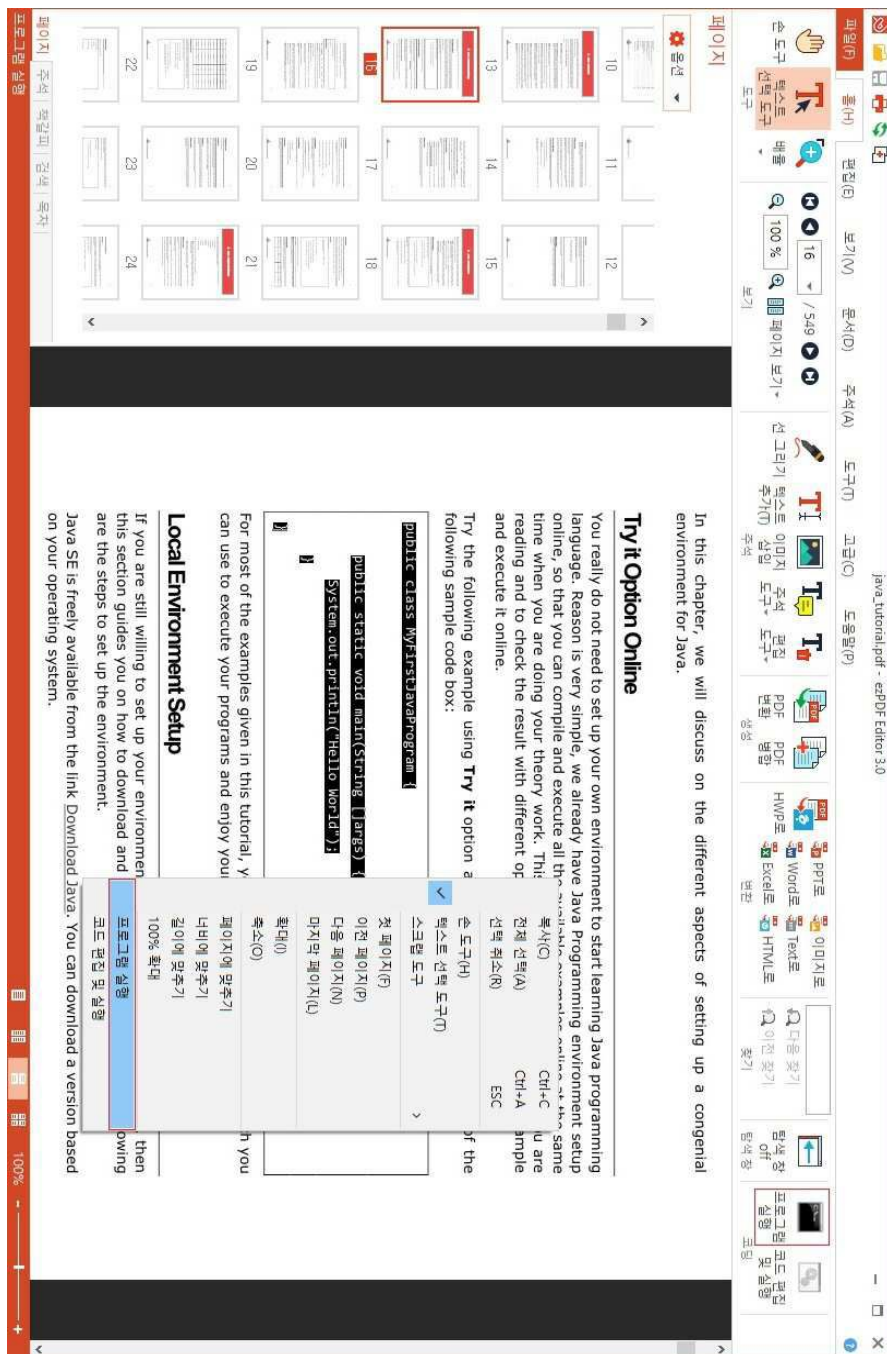
도면1



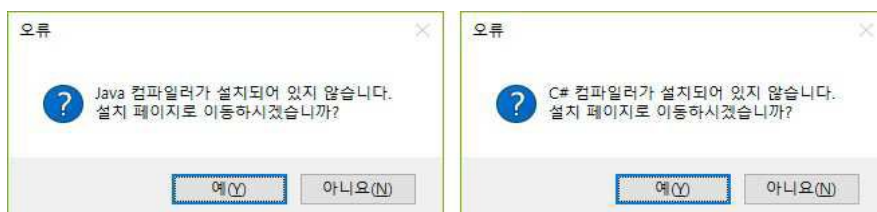
도면2



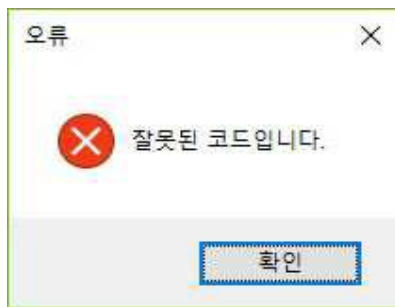
도면3



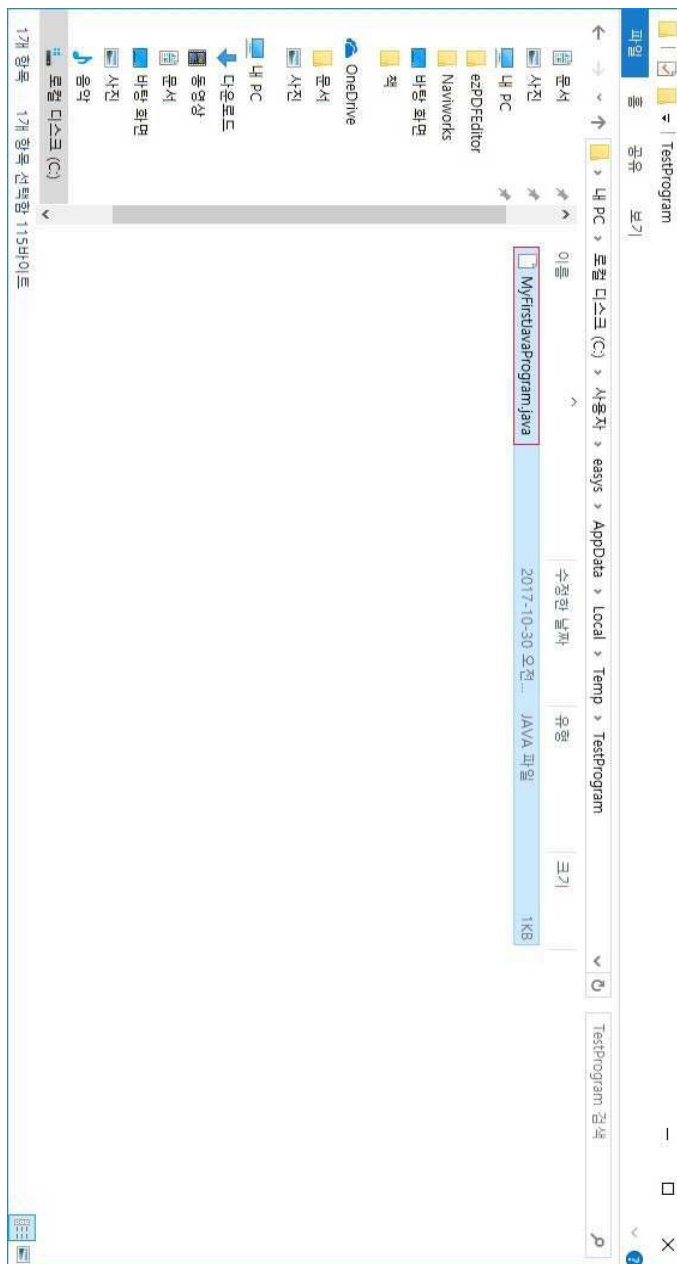
도면4



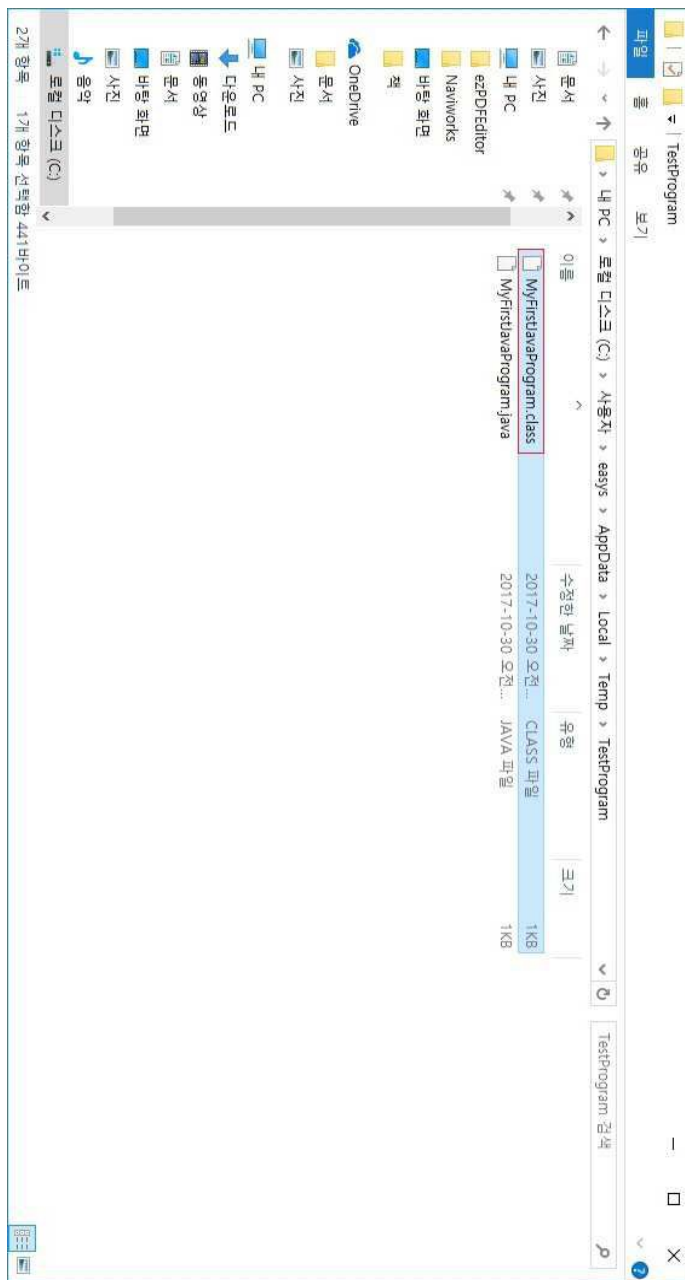
도면5



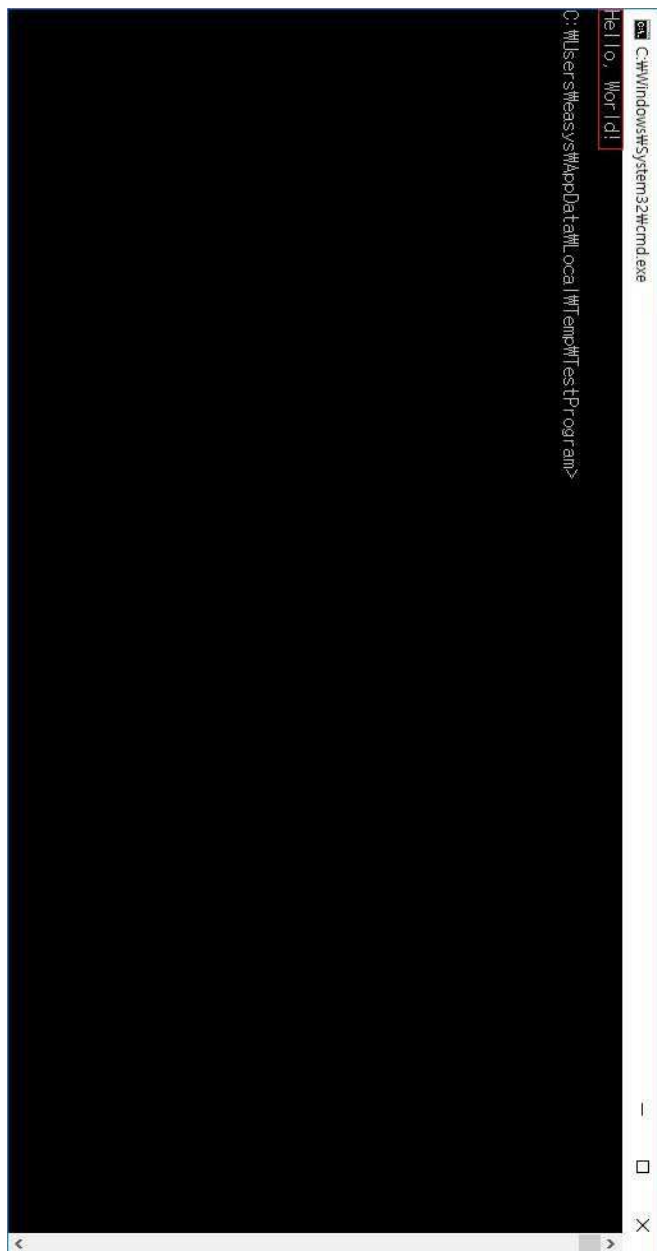
도면6



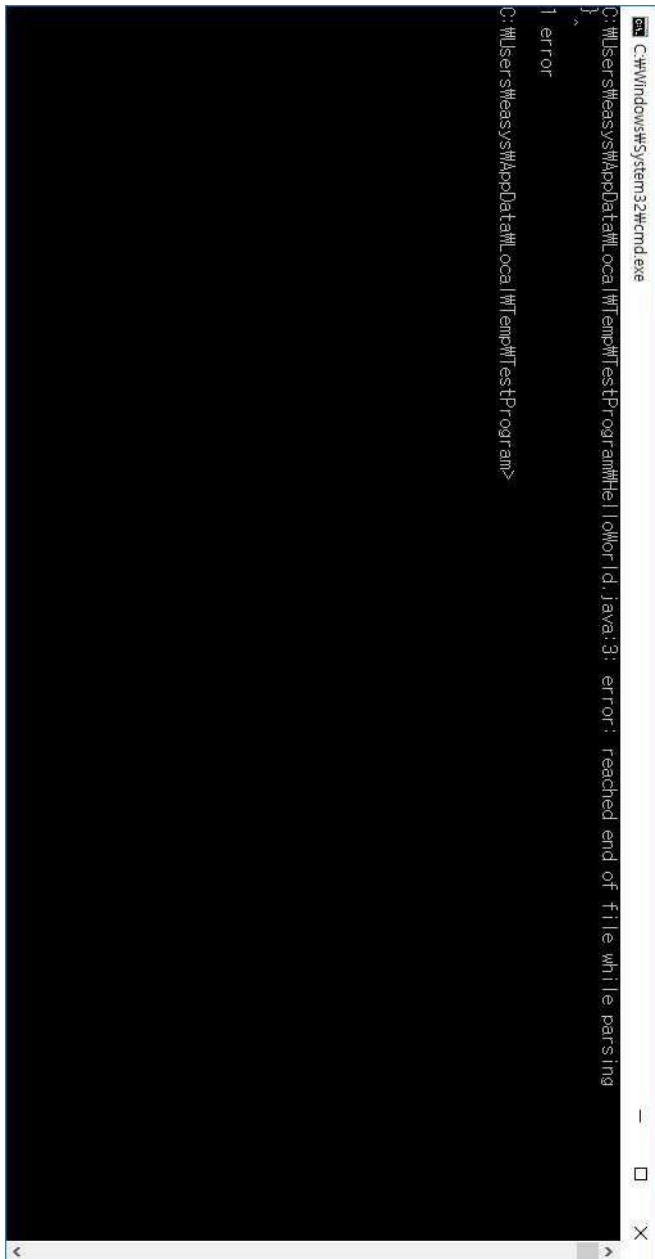
도면7



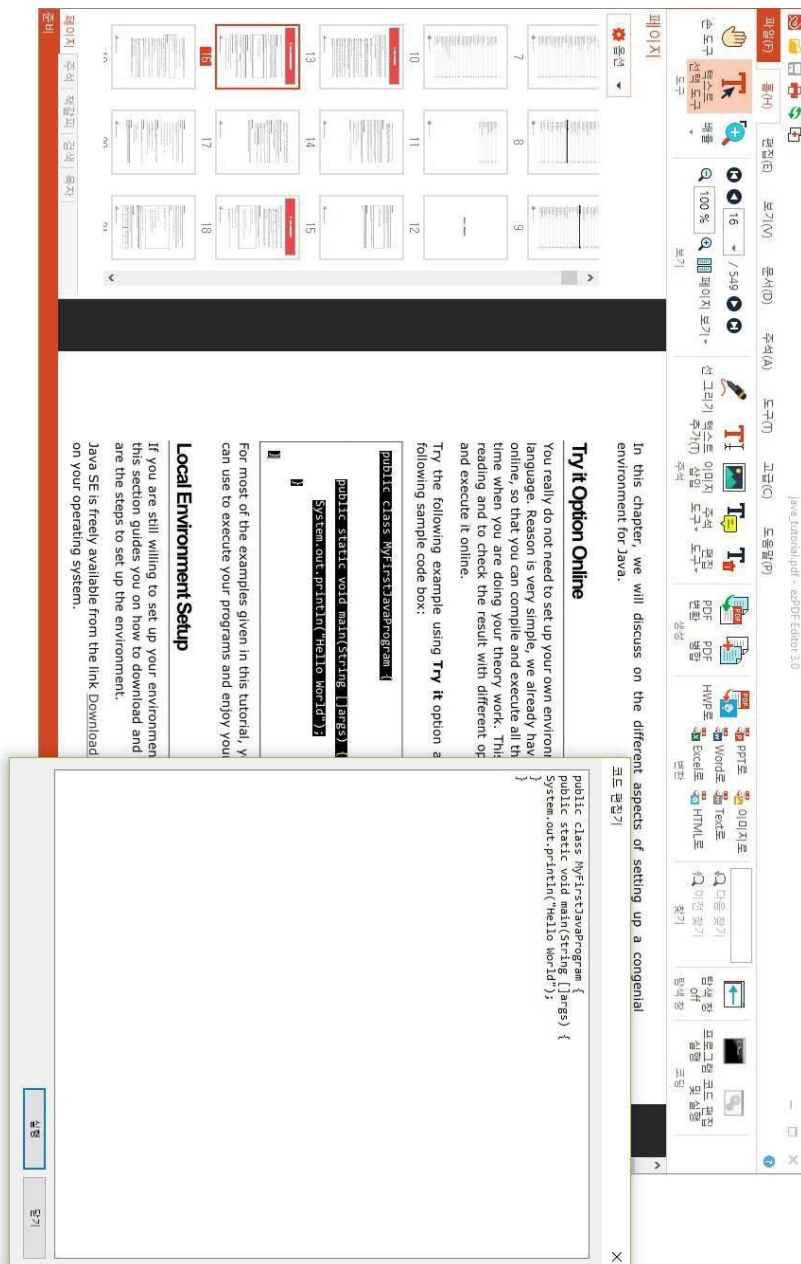
도면8



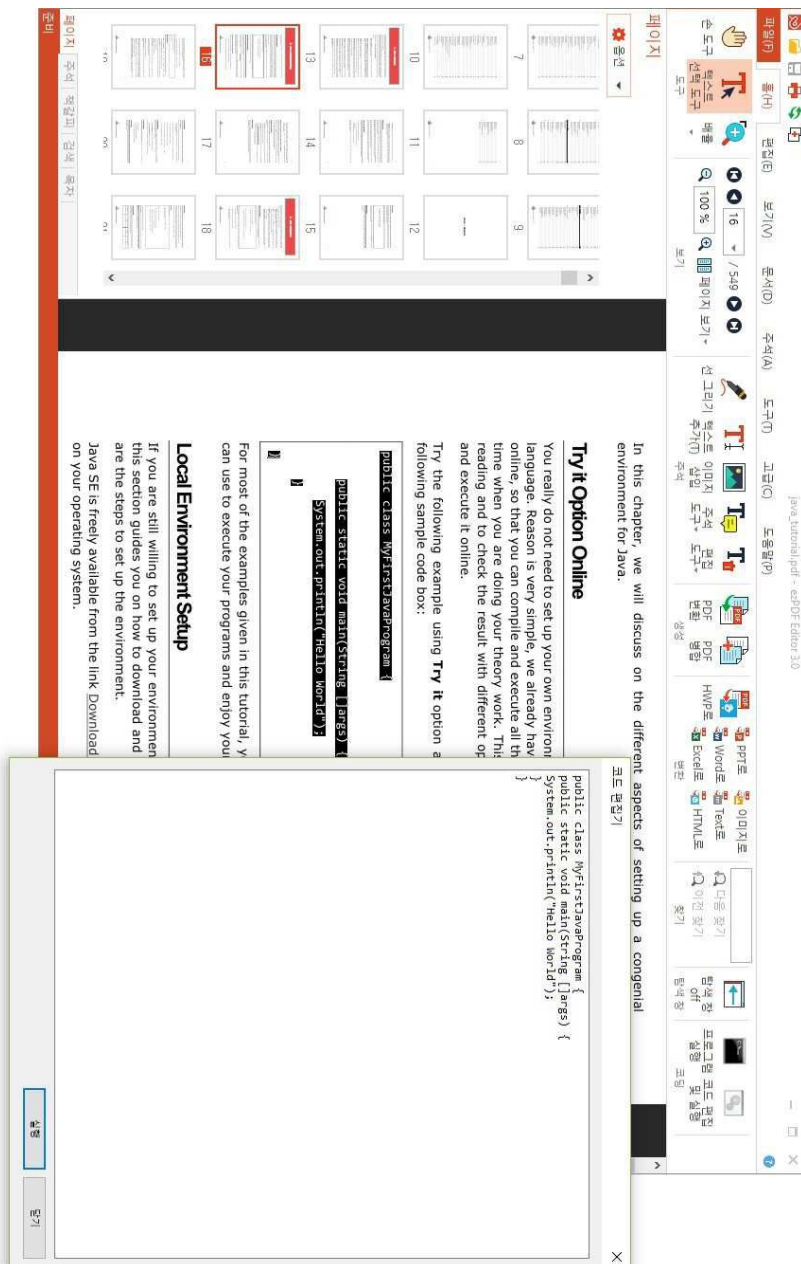
도면9



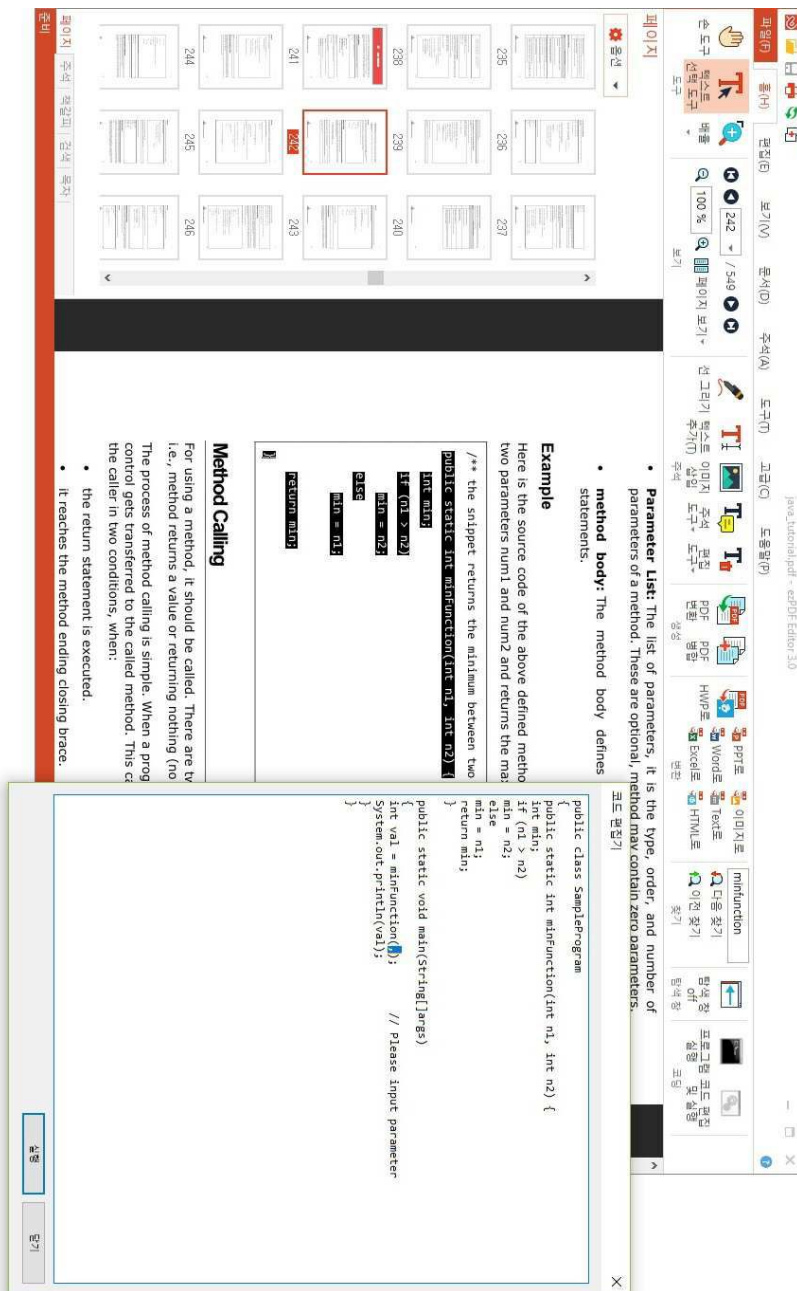
도면10



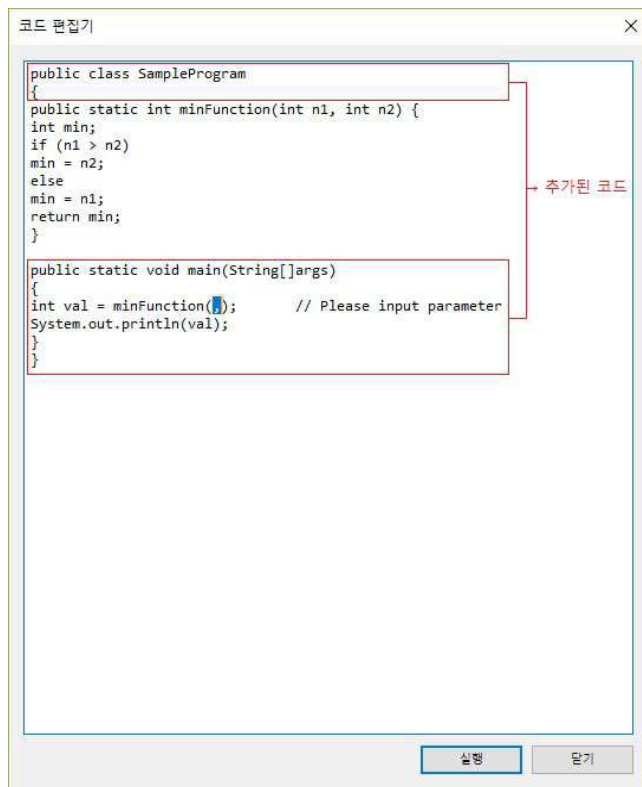
도면11



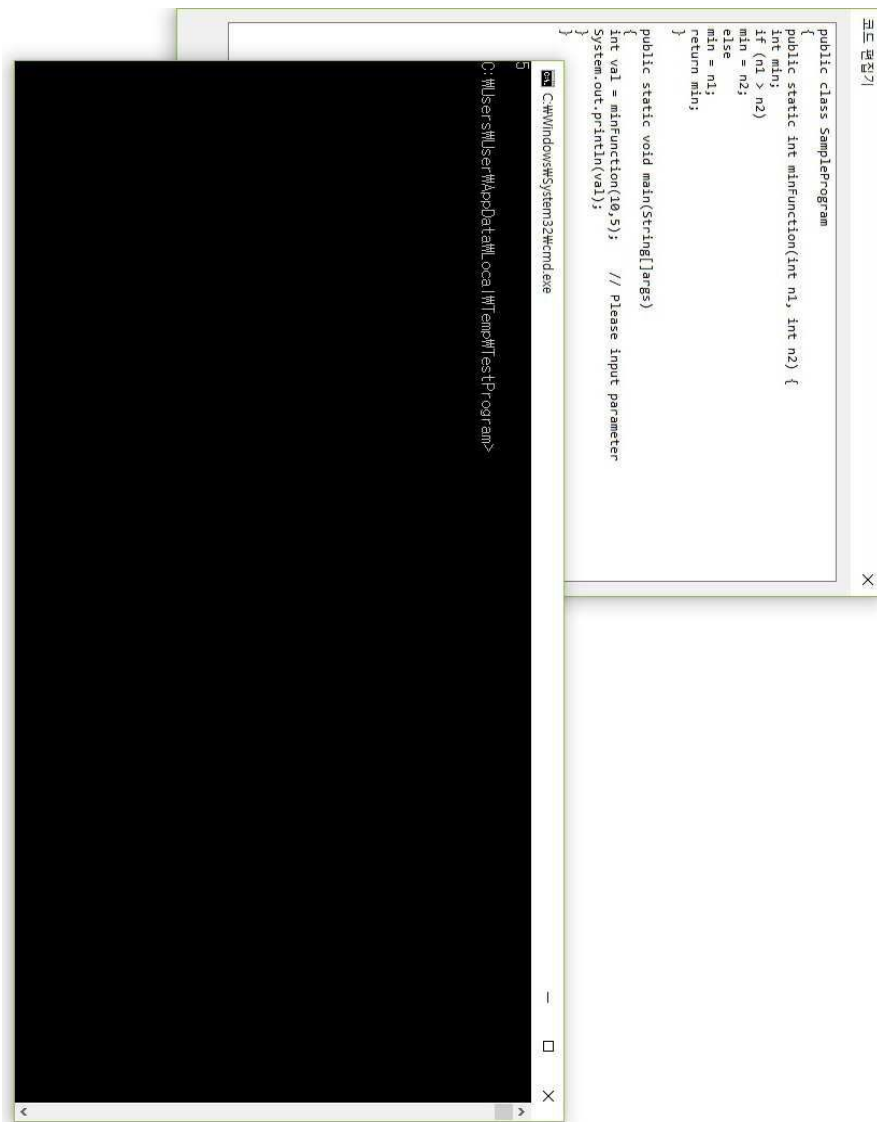
도면12



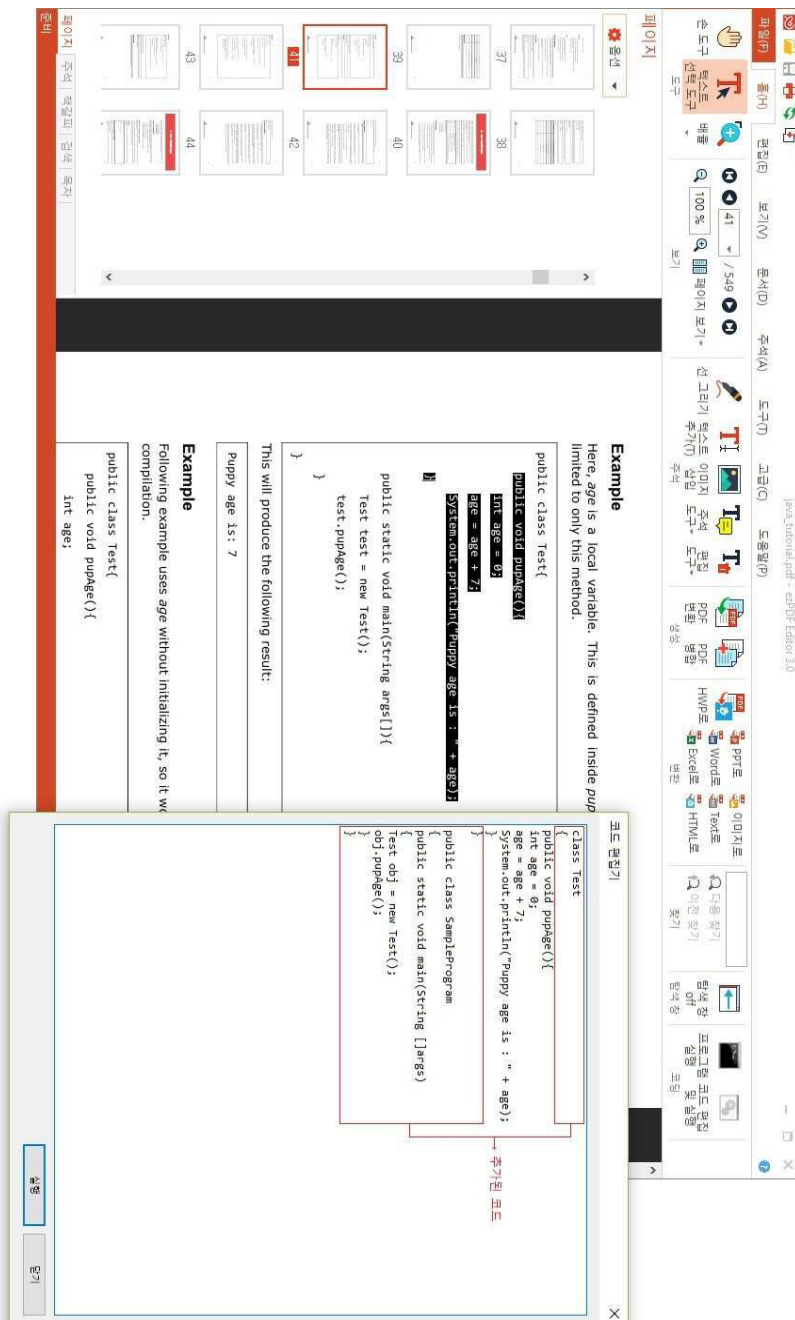
도면13



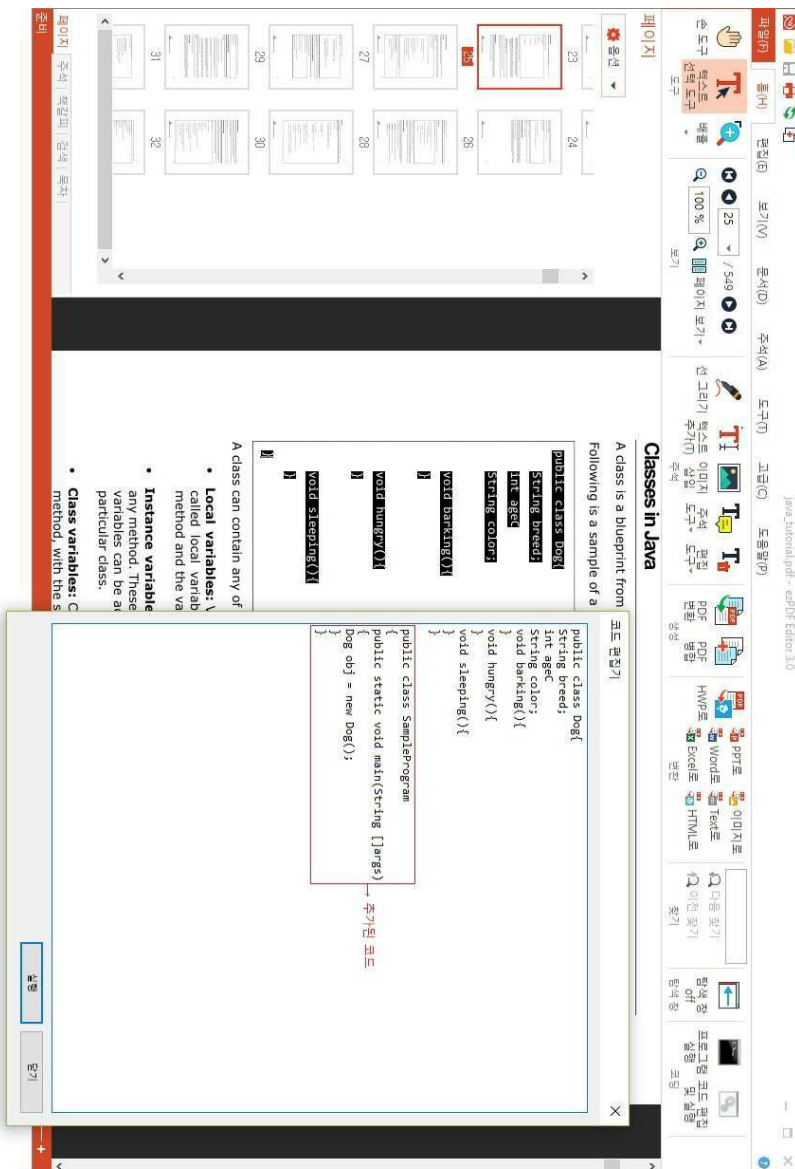
도면14



도면15



도면16



도면17

