

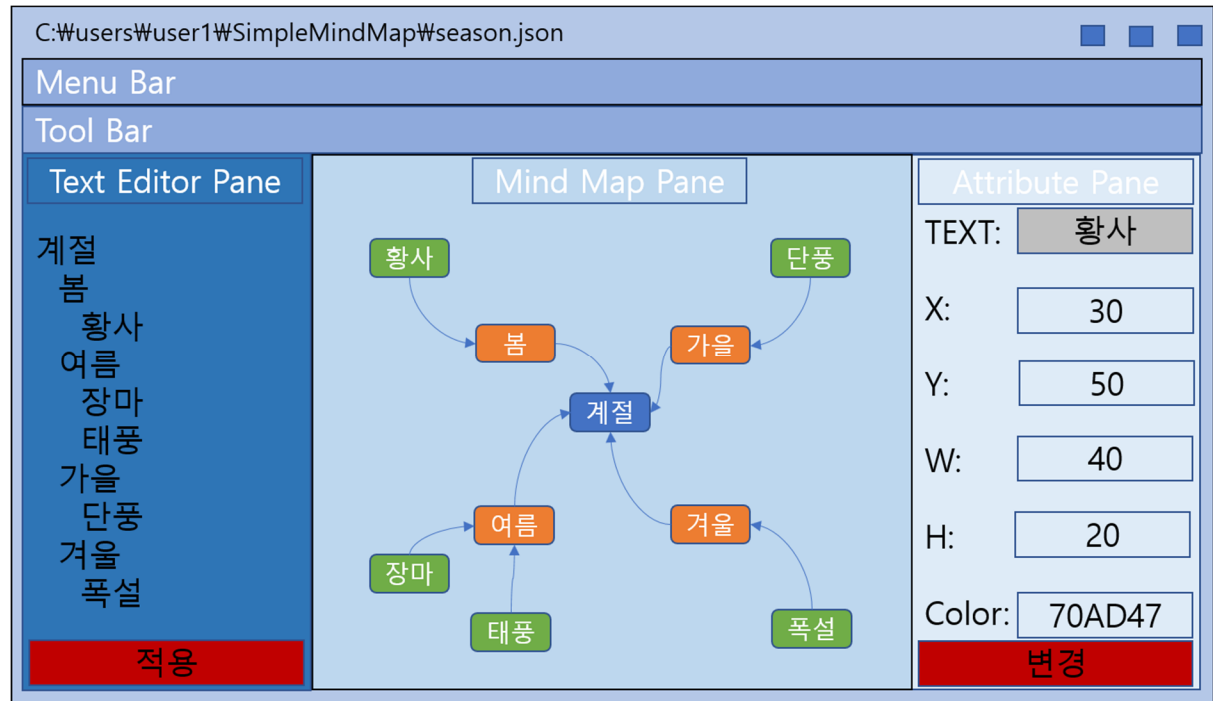
간단한 마인드맵 애플리케이션 요구 사항

데드라인: 6월 14일 23시 59분 59초 (주의: +1일 적용 안함)

제출물: 보고서 + 소스파일 -> 압축 -> (이름_학번)+.zip

1. 사용자 인터페이스 요구 사항

프로그램 실행 시 GUI는 [그림 1]과 같은 구조가 되도록 합니다.



[그림 1] 화면 구성

각 GUI 구성 요소는 [표 1]의 해당 SWING 컴포넌트를 사용하도록 합니다.

GUI 구성 요소	SWING 컴포넌트
윈도우	JFrame
메뉴바	JMenuBar
툴바	JToolBar
속성 페인, 마인드맵 페인	JPanel (in JScrollPane in JSplitPane)
텍스트 에디터 페인	JTextArea (in JScrollPane in JSplitPane)
노드	JLabel

[표 1] GUI 구성 요소를 구현하기 위한 SWING 컴포넌트

2. [텍스트 에디터 페인]의 요구 사항

한 행이 하나의 노드에 대응합니다.

탭으로 레벨을 결정합니다.

'적용' 버튼을 클릭하면 마인드 맵 구조를 계산하여 Mind Map Pane에 가시화 시킵니다.

3. [속성 페인]의 요구 사항

속성 페인은 선택된 노드의 속성을 디스플레이 합니다. 속성 페인에서 보여주어야 할 정보는 [표 2]를 따릅니다. 속성 페인은 JLabel을 사용하여 보여주고자 하는 정보가 무엇인지 표시하며, 속성 값은 JTextField를 사용하여 디스플레이 합니다. JTextField를 사용하는 이유는 사용자의 속성 값 변경을 허용하기 위함입니다.

추가적으로, 속성 페인에 [변경] 버튼(JButton)을 추가하십시오. 이 버튼의 용도는 사용자에 의한 변경된 속성 값을 MindMap Pane에 반영하는 역할을 수행합니다.

속성	SWING 컴포넌트
시작 x, y 좌표	JLabel, JTextField
너비, 높이	JLabel, JTextField
(노드) 텍스트 (수정되지 않도록 설정)	JLabel, JTextField
색상	JLabel, JTextField
'변경' 버튼	JButton

[표 2] 속성 페인의 속성

4. 마인드맵 페인의 요구 사항

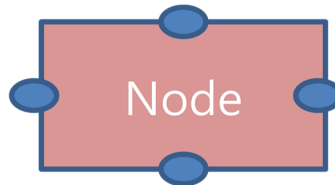
마인드 맵 페인에서 처리해야 할 마우스 이벤트는 [표 3]을 따릅니다.

이벤트	액션
노드의 선택	선택된 노드의 속성은 속성 페인에 표시되며, 노드의 배경색을 반전시킵니다.
노드의 위치 이동	1. 특정 노드에서 마우스를 프레스 한 후, 마우스 드래그를 통하여 마우스 포인트를 이동시킨 상태에서 마우스를 릴리즈 하게되면 그 위치로 노드가 이동되도록 합니다. 2. 노드의 위치 이동 이벤트는 노드 사이의 연결선이 다시 그려지도록 합니다.
노드의 크기 변경	1. 특정 노드의 '상/하/좌/우 모서리' 혹은 '상/하/좌/우 연결점'에서 마우스를 프레스 한 후, 마우스 드래그를 통하여 마우스 포인트를 이동시킨 상태에서 마우스를 릴리즈 하게되면 그 위치로 노드의 모서리가 변경되도록 합니다. 2. 노드의 크기 변경 이벤트는 노드 사이의 연결선이 다시 그려지도록 합니다.

[표 3] 마인드맵 페인에서의 마우스 이벤트

노드 컴포넌트(JLabel)은 마인드맵 페인(JPanel)위에 올려 놓는 방식(add() 메소드 호출)을 취하도록 합니다. 그러나, 노드 사이의 연결선은 마인드맵 페인에서 직접 그리도록 구현합니다.

각 노드는 [그림 2]와 같이 4개의 연결점을 갖습니다. 각 연결점은 연결선의 시작 및 끝에 해당합니다. 마우스 이벤트에 의한 노드 위치 이동 혹은 속성 페인에서 속성 변경에 의한 크기 및 위치가 변경 될 경우 연결선의 시작과 끝 점을 새로이 계산 한 후, 그 결과에 따라 연결선이 다시 그려지도록 구현합니다. 주의, 연결점을 [그림 2]와 같이 화면에 표시되도록 구현하는 것은 아닙니다. 연결선의 시작과 끝 점을 새롭게 계산할 때 기준으로 삼으라는 것입니다.



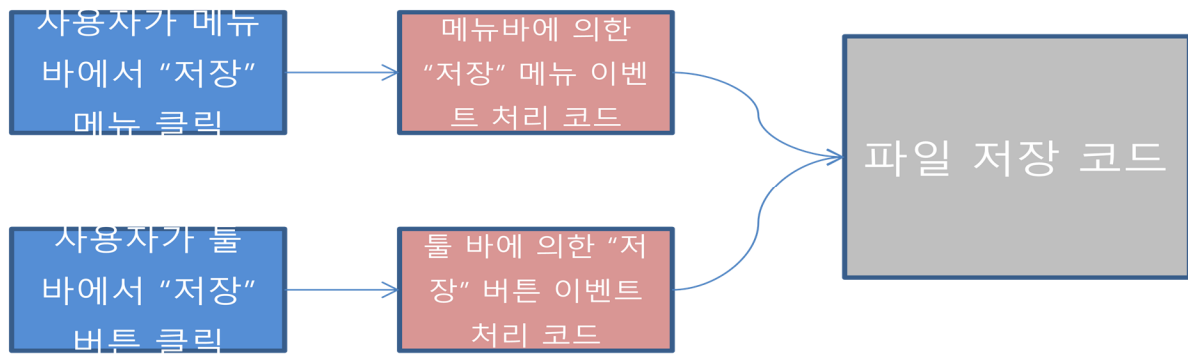
[그림 2] 노드의 연결점

5. 메뉴 바, 툴 바의 요구 사항

메뉴 바와 툴 바에서 처리해야 할 액션은 [표 4]를 참고하세요.

이벤트	액션
새로 만들기	1. 새로운 마인드맵 편집을 시작합니다. 2. 이미 편집 혹은 불러낸 내용이 있다면, 프로그램이 처음 시작 했을 때의 상태로 되돌립니다.
열기	파일 로드
저장	파일 저장
다른 이름으로 저장	새로운 파일 생성
닫기	프로그램 종료
적용	텍스트 편집 내용을 마인드 맵 페인에 적용
변경	속성 페인 변경 내용을 마인드 맵 페인에 적용

모든 이벤트는 메뉴 바와 툴 바 모두에서 발동 가능하도록 구현합니다. 그러나, 각 이벤트에 해당하는 액션은 한 벌만 구현하여 메뉴 바에서 발동 되었건 툴 바에서 발동 되었건 동일한 액션 구현부가 실행 되도록 합니다. [그림 3]을 참고해서 이해하기 바랍니다. [그림 3]은 "저장" 이벤트를 예로 삼은 것입니다.



[그림 3]

6. 파일 생성에 대한 요구 사항

파일 저장은 1. 객체 저장 방식, 2. 여러분이 고안한 파일 스펙에 의거하여 바이너리 혹은 문자 방식 스트림을 사용하여 저장 하는 방식, 3. XML 혹은 json 타입으로 저장하는 방식의 세 가지 중 하나를 선택합니다. 3의 경우 라이브러리를 이용하여 save/load 기능을 구현합니다. 라이브러리는 많이 사용되는 것을 선택하는 것이 향후(이번학기 이후에도)에 도움이 될 것입니다.

배점은 $3 > 2 > 1$ 순으로 높은 점수를 받게 될 것입니다.

7. 마인드맵 모델에 대한 요구 사항

마인드맵 페인에서 일어나는 에디팅 상황을 보관 할 자료 구조는 "트리 모델"을 사용하기 바랍니다. 이 트리의 각 노드는 마인드맵 노드(JLabel) 하나를 의미하며 따라서, 이 노드에는 속성 페인에 디스플레이 할 내용을 모두 담고 있습니다. 또한 이벤트에 의해 노드의 속성이 변경되면 이 트리 노드 또한 정보가 갱신됩니다. 마지막으로 파일 저장은 이 마인드맵 모델의 정보만을 저장 하면 됩니다. 물론, "열기"와 같은 이벤트에 의해서는 파일에서 불러 온 내용을 기반으로 이 트리 모델을 복원하면 됩니다. 이해가 잘 안가는 학생은 "MVC 모델"이라는 것을 학습하세요. 이 절의 "마인드맵 모델"은 MVC에서 M에 해당하는 역할을 수행합니다. 참고로, "마인드맵 모델"은 POJO 형식을 띄게 되어 있으며 이 모델만 파일 저장시 저장하면 되며, 파일로드 또한 이 모델만 불러 와서 다시 객체화하면 됩니다. 오해하지 말 것은 MVC 모델대로 구현하라는 것이 아닙니다. 단지 MVC 모델처럼 역할을 모듈로 나누어 상황에 맞게 구현하시면 됩니다.

8. 그 외의 요구 사항

지금까지 설명한 요구 사항을 기반으로 자유롭게 구현하세요. 상기한 요구 사항은 최소 요구사항으로 받아들이세요. 요구하지 않은 기능 이상 혹은 언급되지 않은 동작에 대한 세부적인 동작 방식은 합리적으로 결정하여 자유롭게 만드세요. 프로젝트 검사 시, 자신이 구현한 사항을 효과적으로 설명할 수 있도록 보고서를 작성해 오세요. 이 보고서는 형식적으로 작성하지 말고, 검사

시 설명을 도울 수 있는 도면, 자신이 요구 사항 이상으로 구현한 내용 목록과 방법 등등 자신이 수준 높게 구현했음을 어필하고자 하는 내용을 넣으시면 되겠습니다.

작업한 내용별 담당자가 누구인지, 얼마의 기간동안 작업했는지를 표로 작성하여 첨부해 주세요. 이것은 프로젝트를 수행하면서 지속적으로 수행내용을 그때그때 기록 및 동기화 해 두셔야 보고서 마감일에 맞추어 최종본을 만들기에 수월할 것입니다.