**EPL Players’ Status**

**Final Project Report**

**19011624 송규민**

* **구현 정도 및 진행 상황**

1. **Load 구현 완료 (xsd validation 검사 포함)**
2. **Make 구현 완료**
3. **Find 구현 완료**
4. **Save 구현 완료**
5. **Print 구현 완료**
6. **Insert 구현 완료**
7. **Update 구현 완료**
8. **Delete 구현 완료**
9. **Exit 구현 완료**

* **중간 제출 이후 수정 사항 요약**

1. **Find() 매커니즘 수정**
2. **print() 매커니즘 수정**
3. **delete() 속성 삭제 기능 추가**
4. **update() 구현 완료**
5. **불필요한 코드 삭제 및 코드 정리**

* **매뉴얼(실행 방법)**

1. **제 컴퓨터 환경에서 jar 파일을 javac 등을 이용한 executable 파일 생성이 불가능합니다. 따라서 소스 코드를 Eclipse에 Import 해야 합니다. 코드를 직접 Eclipse 에디터에 import 한 후 Package명만 바꿔서 실행하면 됩니다.**
2. **작동에 사용되는 코드는 총 두 가지입니다(MainWindow.java, XMLFunctions.java)**
3. **자바 src 폴더가 존재하는 위치에 xml 파일 및 xsd 파일이 존재해야 합니다.**
4. **이클립스 환경에서 실행 시 라이브러리 import 과정에서 오류 발생 시 프로젝트 폴더 우클릭 -> Build Path -> Confiure Build Path -> Libraries -> Add External Jars 로 Xerxes jar 파일들을 import 합니다.**
5. **이클립스 환경에서 실행 시 jdk 버전을 8 이하로 설정해야 정상 동작합니다.**

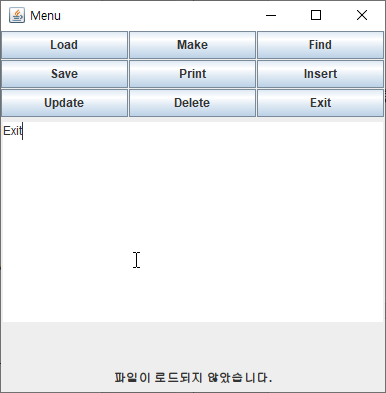
* **기능 설명**

1. **Load** 
   1. **파일을 Java 프로그램에 적재합니다.**
2. **Make**
   1. **빈 xml 파일을 기본 경로에 생성한 뒤 마찬가지로 적재합니다.**
   2. **빈 xml 파일의 이름을 정할 수 있습니다.**
3. **Find**
   1. **트리 형식으로 구성된 콤보박스에서 Element, Comment, Text 등의 노드들을 찾습니다.**
   2. **찾으면 부모 노드 정보, 노드 타입, 노트 이름, 노드 값, 자식 정보 등이 나타납니다.**
4. **Save**
   1. **현재까지 변경된 사항을 새로운 파일로 저장합니다.**
   2. **새로운 파일의 이름을 정할 수 있습니다.**
5. **Print**
   1. **현재까지 변경된 DOM 트리를 출력합니다.**
6. **Insert**
   1. **새로운 다양한 노드(Element, Comment 등)을 삽입합니다**
7. **Update**
   1. **현재 존재하는 노드들의 값이나 이름을 수정합니다.**
8. **Delete**
   1. **현재 존재하는 노드들을 통째로 삭제합니다.**
9. **Exit**
   1. **프로그램을 종료합니다.**

* **기능 분석 및 시나리오**

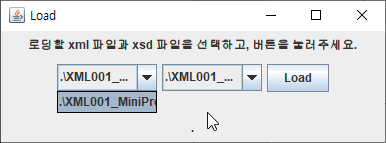
1. **Load**
   1. **왼쪽 콤보박스에서 xml 파일을 선택할 수 있습니다. 그리고 오른쪽 콤보박스에서 xml 파일에 대한 스키마 파일(xsd)을 선택할 수 있습니다. 왼쪽 콤보박스에서는 xml 파일만 선택할 수 있고, 오른쪽 콤보박스에서는 xsd 파일만 선택할 수 있습니다.**
   2. **xml파일과 xsd파일을 선택한 뒤 load 버튼을 클릭하면 loading이 완료됩니다. loading 하는 과정에서 xsd validation을 시행합니다. validation을 통과하면 창이 닫히며 적재가 완료되고 실패하면 실패 메시지가 등장합니다.**
   3. **Load를 제일 먼저 실행해주어야 다른 기능들을 사용 가능합니다. (혹은 Make를 제일 먼저 실행해주어야 합니다.) 이를 지키지 않으면 경고 문구가 뜨며 기능이 실행되지 않습니다.**
2. **Make**
   1. **클릭 시 생성할 파일명을 물어보는 창이 뜹니다. xml 확장자에 맞게 파일명을 입력하면 즉시 빈 xml 파일이 생성됩니다. 어떤 노드도 존재하지 않습니다.**
   2. **Load와 더불어 다른 기능을 사용하기 위해 제일 먼저 실행해주어야 하는 기능입니다.**
3. **Find**
   1. **키워드를 Combobox를 이용해 찾아서 그 노드에 대한 정보를 탐색하는 기능입니다. Combobox에서는 Element와 Text, Comment로 구성되어 있습니다.**
   2. **모든 Element에 기본적으로 달려 있는 Text 중 의미가 없는 공백 혹은 줄바꿈으로 이루어진 Text는 검색 시에 Combobox에 나타나지 않도록 설정했습니다. 즉, Text는 문구가 담겨 있는 Text만 검색 트리에 나타납니다.**
   3. **Text를 검색 트리에서 찾아볼 때 자신의 부모 노드, 즉 태그의 이름을 가지고 있어 어디에 속한 Text인지 구분하기 쉽게 했습니다.**
   4. **노드를 선택한 뒤 FInd 버튼을 클릭하면 Message 창이 뜹니다. Message 창에는 선택한 노드의 타입, 노드의 이름, 노드 안에 담겨 있는 값, 노드의 부모 정보, 노드의 자식 개수가 나타납니다.**
4. **save**
   1. **로드된 파일에 대해 현재까지 삽입, 수정, 삭제 수행에 대한 변경 결과를 xml 형식대로 저장합니다. 파일명을 원하는대로 지정할 수 있습니다.**
5. **print**
   1. **로드된 파일에 대해 현재까지 삽입, 수정, 삭제 수행에 대한 변경 결과를 출력합니다. ‘ㄴ’자 구조로 되어 있어 어떤 노드가 어떤 노드의 자식인지 부모인지, 자식의 개수가 몇 개이고, 깊이는 어느 정도 되는지 한 눈에 파악 가능합니다. FInd와 마찬가지로 Text는 의미가 있는 Text 노드만 출력했습니다.**
6. **Insert**
   1. **노드를 추가합니다. 노드를 추가할 때 노드 타입과 노드 위치를 지정해 주어야 합니다. 노드타입은 Element, Text, Attribute, Comment 중 하나를 선택해서 생성할 수 있습니다. 노드 위치를 검색할 때는 Element 노드만 선택할 수 있습니다.**
   2. **Element를 생성할 시에 기준으로 지정한 위치 이전에 생성합니다. 즉 기준으로 지정한 노드보다 한 차례 앞선 형제가 됩니다. Text와 Comment, Attribute는 기준 노드의 자식으로 생성됩니다.**
   3. **Element와 노드의 이름을 입력합니다. 그리고 Text와 Comment는 노드의 내용물을 입력합니다. 마지막으로 Attribute는 InputMessage 창을 이용해 두 번 입력을 받습니다. 첫 번째 입력은 속성의 이름, 두 번째 이름은 속성의 값입니다.**
   4. **최상위 노드에는 삽입이 불필요하기 때문에 최상위 노드에서의 삽입 작업을 할 수 없도록 설정해 놓았습니다. 최상위 노드에서 삽입을 시도할 시 경고 메시지가 뜹니다.**
7. **Update**
   1. **노드를 수정합니다. Insert와 마찬가지로 노드 타입을 선택한 후 그 노드(Element) 혹은 그 노드가 속해 았는 부모 노드(Text, Attribute, Comment)를 선택해야 합니다. 따라서, 노드 위치를 검색할 때는 Element 노드만 검색할 수 있습니다.**
   2. **기준 Element를 선택하고 Update 버튼을 클릭하면, 그 Element명을 바로 수정하거나, 그 Element 내에 들어있는 지정된 타입의 내용물을 선택해 수정할 수 있습니다.**
   3. **Attribute를 수정할 때는 Attribute 명과 Attribute 내용을 모두 입력해야 합니다. 다른 타입은 한 개만 입력하면 됩니다.**
8. **Delete**
   1. **노드를 삭제합니다. Insert와 마찬가지로 노드 타입을 선택한 후 그 노드(Element) 혹은 그 노드가 속해 았는 부모 노드(Text, Attribute, Comment)를 선택해야 합니다. 따라서, 노드 위치를 검색할 때는 Element 노드만 검색할 수 있습니다.**
   2. **Insert와 다르게 최상위 노드를 삭제할 수 있습니다.**
   3. **기준 Element를 선택하고 Delete를 클릭하면 Element 타입을 삭제 시에는 바로 자식 요소들과 함께 사라지고, 다른 타입을 삭제 시에는 삭제할 자식 요소들(혹은 Attribute 요소들)을 선택하면 삭제할 수 있습니다.**
9. **Exit**
   1. **프로젝트를 탈출합니다. 탈출 시 Save 기능이 자동으로 실행됩니다. Save 기능과 동일하게 작동합니다.**

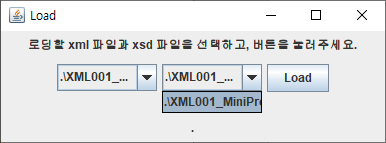
* **시나리오 별 GUI 화면**
  + **시나리오**
    - **프로젝트를 킵니다.**
    - **파일이 경로명에 있는 것을 확인한 뒤 xml 파일과 xsd 파일을 불러옵니다.**
    - **불러온 파일의 트리를 확인합니다.**
    - **파일에 newElement 요소와 거기에 “newText” 문장을 추가합니다.**
    - **newElement 요소에 isTest 키와 true 값을 가진 Attribute를 추가합니다.**
    - **h2를 삭제합니다.**
    - **picture의 alt 속성을 삭제합니다.**
    - **newElement 요소명을 oldElement로 수정합니다.**
    - **다시 파일의 트리를 출력합니다.**
    - **exit를 눌러 파일을 “new.xml”로 저장하고 종료합니다.**
  + **GUI 화면**
* **프로젝트를 킵니다.**
* **파일이 경로명에 있는 것을 확인한 뒤 xml 파일과 xsd 파일을 불러옵니다.**
* **불러온 파일의 트리를 확인합니다.**
* **파일에 newElement 요소와 거기에 “newText” 문장을 추가합니다.**
* **newElement 요소에 isTest 키와 true 값을 가진 Attribute를 추가합니다.**
* **h2를 삭제합니다.**
* **picture의 alt 속성을 삭제합니다.**
* **newElement 요소명을 oldElement로 수정합니다.**
* **다시 파일의 트리를 출력합니다.**
* **exit를 눌러 파일을 “new.xml”로 저장하고 종료합니다.**
* **프로젝트를 킵니다.**

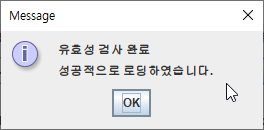
****

**첫 실행화면입니다. Load를 먼저 실행합니다.**

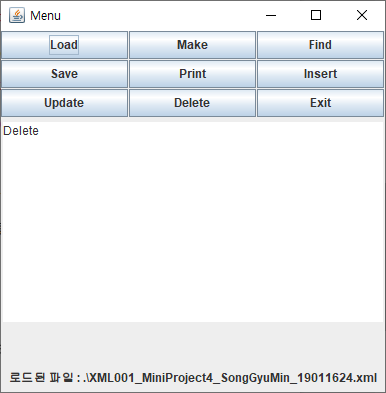
* **파일이 경로명에 있는 것을 확인한 뒤 xml 파일과 xsd 파일을 불러옵니다.**

****

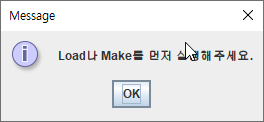
****

****

**xml 파일과 xsd 파일을 선택한 후 Load를 클릭하면 유효성 검사와 동시에 종료됩니다. 그리고 다시 메인 화면으로 돌아갑니다.**

****

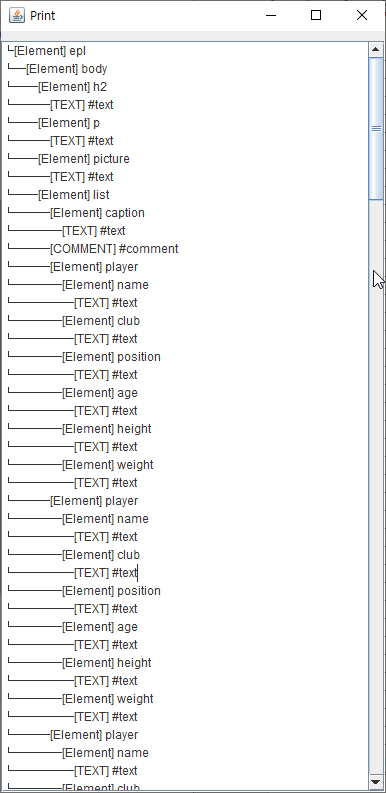
**파일을 로드한 후에는 JFrame 하단에 현재 로드되어 있는 파일이 보입니다.**

****

**만약 Load나 Make를 먼저 실행하지 않으면 다음과 같은 경고 문구가**

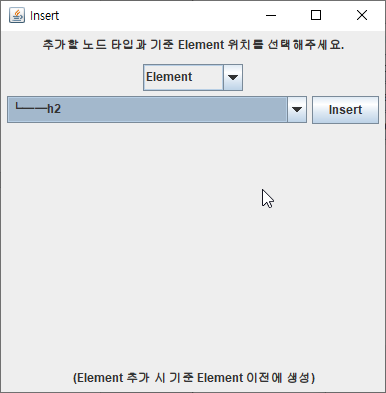
**뜹니다.**

* **불러온 파일의 트리를 확인합니다.**

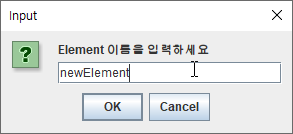
****

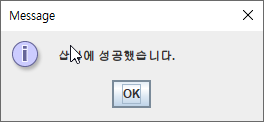
**다음은 Print로 현재 로드된 파일을 출력했을 때 모습입니다.**

* **파일에 newElement 요소와 거기에 “newText” 문장을 추가합니다.**

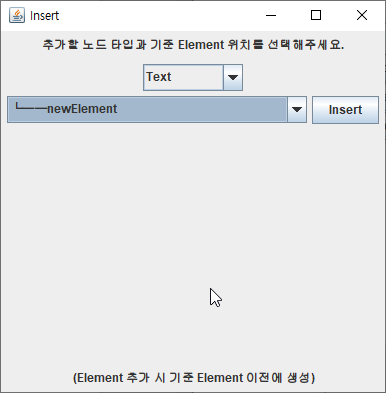


**h2 요소 위에 생성하겠습니다.**

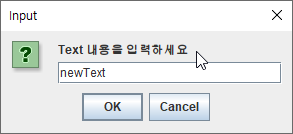




**요소를 생성합니다.**



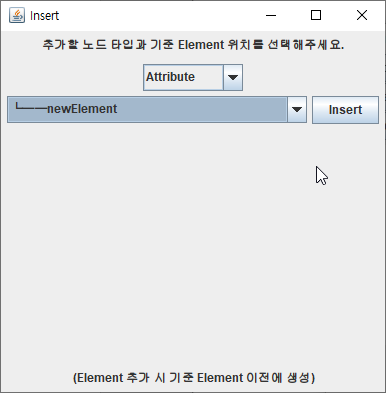
**이어서 그 요소에 Text를 추가합니다.**



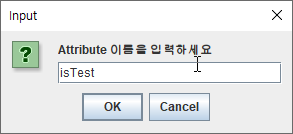


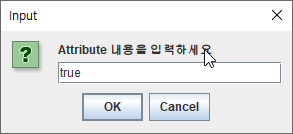
**텍스트 삽입에 성공했습니다.**

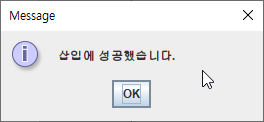
* **newElement 요소에 isTest 키와 true 값을 가진 Attribute를 추가합니다.**



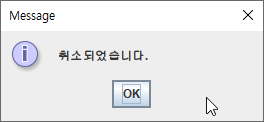
**타입을 속성으로 설정하고 속성을 삽입할 요소를 선택합니다.**





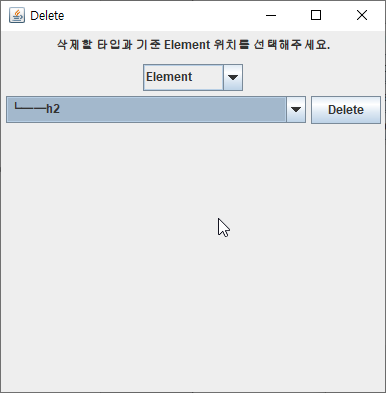


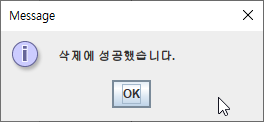
**속성 삽입에 성공했습니다.**



**추가로 어느 타입을 생성하든지 중간에 입력을 그만두면 위와 같은 취소 창이 뜨게 됩니다.**

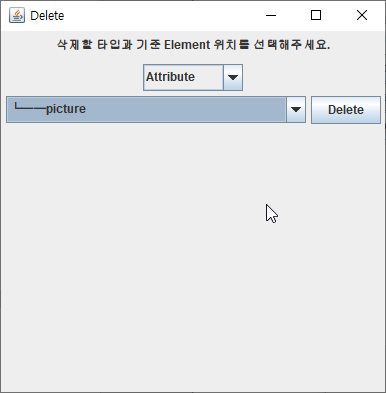
* **h2를 삭제합니다.**





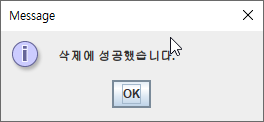
**자세한 설명은 생략하겠습니다. 삽입 시와 유사한 삭제 화면입니다.**

* **picture의 alt 속성을 삭제합니다.**



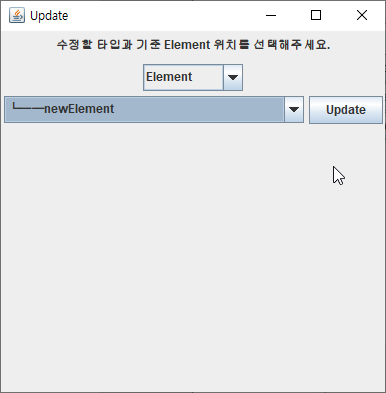
텍스트이(가) 표시된 사진

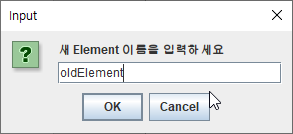
자동 생성된 설명

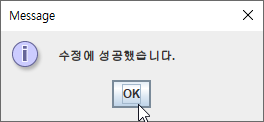


**속성 삭제를 완료했습니다.**

* **newElement 요소명을 oldElement로 수정합니다.**

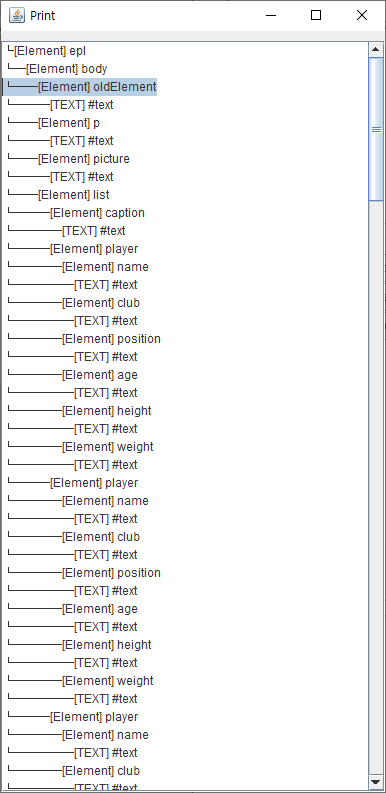






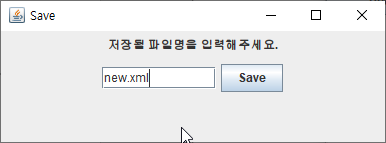
**요소명을 수정했습니다.**

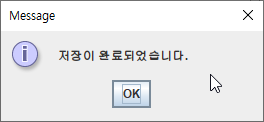
* **다시 파일의 트리를 출력합니다.**



**초기와 다르게 h2 속성이 사라지고, oldElement와 Text가 추가된 것을 확인할 수 있습니다.**

* **exit를 눌러 파일을 “new.xml”로 저장하고 종료합니다.**





**저장이 완료되었습니다.**

텍스트이(가) 표시된 사진

자동 생성된 설명

**new.xml 파일의 내용입니다. 모든 변경사항이 제대로 적용된 것을 확인할 수 있습니다.**

**-끝-**