고급소프트웨어실습I

7주차 보고서

20171646 박태윤

실습에 존재하는 주요 컴포넌트 들은 DeletedList, DeletedTodoItem, DoneList, TodoInsert, TodoItem, TodoTemplate, WhatTodo가 존재한다.

- DeletedList

TodoList프로그램에서 delete된 리스트 항목들을 보여주는 컴포넌트입니다. DeletedList함수에서 삭제된 항목들을 나타내는 배열 deletedTodos를 인자로 받아 TODO LIST창의 DELETED LIST에 리스트로 항목들을 보여줍니다.

- DeletedTodoItem

삭제된 항목 하나를 나타내는 컴포넌트입니다. 앞서 설명한 DeletedList컴포넌트에서 이 DeletedTodoItem컴포넌트를 이용하여 TODO LIST창의 DELETED LIST에 표현이 되어 있는 리스트 한 줄을 표현합니다.

- DoneList

TODO LIST창에서 체크가 된(수행이 완료된 일) 리스트 항목들을 보여주는 컴포넌트입니다. DoneList함수에서 완료된 항목들을 나타내는 배열 doneTodos에 대해 TodoItem컴포넌트를 이용하여 객체를 하나 생성해서 TodoList프로그램에서 수행이 완료된 일들을 표현하는 TODO LIST창의 DONE LIST에 항목들을 리스트로 보여주는 컴포넌트입니다.

- WhatTodo

TODO LIST창에서 해야 할 일에 대한 항목들을 보여주는 컴포넌트입니다. 해야 하는 일의 이름을 TodoInsert컴포넌트를 통해 입력을 받아 해야 할 일들을 나타내는 배열 todo에 대해 TodoItem컴포넌트를 이용하여 객체를 생성해 TODO LIST창의 WHAT TO DO에 해야 하는 일들에 대한 항목들을 리스트로 보여줍니다.

- TodoItem

TodoList프로그램에서 일 하나를 표현하는 컴포넌트입니다. todo.checked값에 따라 해야 하는 일인지, 수행이 완료된 일인지를 나타낼 수 있으며 체크가 되어 있다면 수행이 완료된 일이기 때문에 줄 표시를 하여 완료되었음을 표시합니다. DoneList에서 이를 이용하여 완료된 일에 대한 객체를 표현하며 WhatTodo에서도 역시 이를 이용하여 해야 하는 일에 대한 객체를 표현합니다.

- TodoInsert

해야 할 일에 대한 입력 창과 엔터 버튼에 대한 컴포넌트입니다. TodoList프로그램에서 일을 입력하고 엔터 키를 누르면 submit을 할 수 있는데 이에 대한 동작들을 표현합니다.

- TodoTemplate

위의 컴포넌트들을 감싸는 전체적인 동작에 대한 컴포넌트입니다. 해야 할 일을 나타내는 todos배열에서 delete해야 하는 객체를 DeletedList에 추가 후 What To do에서 제외를 하거나 수행이 완료된 일을 나타내는 DoneList에서 동작에 따라 DeletedList에 일부 객체를 추가한 이후 Done List에서 제외를 하는 동작 등을 수행합니다. 실습에서는 DoneList에서 DeletedList로 옮기는 동작, What To Do에서 Done List로 옮기는 동작, Done List에서 다시 What To Do로 옮기는 동작에 대한 코드를 구현하였습니다.

실습에서는 onRemoveDoneTodo, onToggleToDo, onToggleDoneToDo함수를 구현해야 합니다.

- OnRemoveDoneTodo

// 주어진 Done List 항목(체크된 항목)의 id로 Done List에서 해당 항목을 찾고 Deleted List에 추가한 후에, 기존 Done List에서 제외

const onRemoveDoneTodo = (id) => {

// 7주차 실습 구현

const deletedDone = doneTodos.find((todo) => todo.id === id);

setDeletedTodos([...deletedTodos, deletedDone]);

setDoneTodos(doneTodos.filter((todo) => todo.id !== id));

};

Done List에 있는 일들 중 delete하고자 하는 일이 있을 때 호출하는 함수입니다. 삭제를 하고자 하는 일의 id를 인자로 받아 이를 doneTodos배열에서 찾아 deletedDone변수에 assign하고 이 deletedDone이 나타내는 일을 setDeletedTodos([…deletedTodos, deletedDone])를 통해 Done List에서 Deleted List로 옮긴 다음 setDoneTodos(doneTodos.filter((todo) => todo.id !== id))를 통해 해당 id의 일을 Done List에서 삭제합니다.

- onToggleToDo

// 주어진 What To Do 항목(체크안된 항목)을 찾아 항목 객체의 속성(키)을 사용하여 체크한 후, Done List에 추가하고 What To Do에서 제외

const onToggleToDo = (id) => {

// 7주차 실습 구현

let switchTodo = todos.find((todo) => todo.id === id);

switchTodo.checked = true;

setDoneTodos([...doneTodos, switchTodo]);

setTodos(todos.filter((todo) => todo.id !== id));

};

Todo List에 있는 사용자가 추가한 일 중에서 Done List에 추가를 하고자 하는 일이 있을 때 호출하는 함수입니다. 변경을 하고자 하는 일의 id를 인자로 받아 todos배열에서 해당 일을 찾고 switchTodo변수에 assign한 뒤 완료된 일이기 때문에 해당 객체의 checked변수 값을 true로 바꿔줍니다. 이후 setDoneTodos([…doneTodos, switchTodo])를 통해 DoneList로 옮긴 뒤 setTodos(todos.filter((todo) => todo.id !== id))를 통해 Todo List에서 해당 일을 삭제합니다.

- onToggleDoneToDo

// 주어진 Done List 안 항목을 찾아 항목 객체의 속성을 사용하여 체크 해제한 후, What To Do에 추가하고 Done List에서 제외

const onToggleDoneToDo = (id) => {

// 7주차 실습 구현

let switchDone = doneTodos.find((todo) => todo.id === id);

switchDone.checked = false;

setTodos([...todos, switchDone]);

setDoneTodos(doneTodos.filter((todo) => todo.id !== id));

};

onToggleToDo와는 반대로 DoneList에서 TodoList로 일을 옮기고자 할 때 호출이 되는 함수입니다. 변경을 하고자 하는 일에 대한 id를 인자로 받아 doneTodos배열에서 이를 찾아 switchDone에 assign한 뒤 체크 표시를 해제시켜야 하므로 checked값을 false로 바꾸고 TodoList로 옮긴 뒤 DoneList에서 삭제합니다.

추가적으로 TodoTemplate에는 WhatToDo에서 DeletedList로 옮기는 onRemoveTodo함수, 작성한 일에 대해 todo객체를 생성하고 todos배열에 저장하는 handleSubmit함수가 구현이 되어 있습니다.

이외에 다음과 같은 함수들이 구현이 되어있습니다.

- TodoInsert 컴포넌트

handleChange : 입력창의 내용이 바뀔 때 마다 호출되는 핸들러입니다. 즉 사용자가 입력을 하면 입력 창의 내용이 입력을 할 때 마다 계속 바뀌는데, 이 함수가 호출이 되면서 해당 내용에 따라 state를 업데이트 시켜줍니다.

hangleKeyPress : 엔터 키를 눌렀을 때 입력한 일의 이름을 submit하는 동작을 나타내는 함수입니다.

handleSubmit : 폼을 제출하는 함수입니다. handleKeyPress함수를 통해 엔터 키를 누르면 입력한 폼을 제출하는데, 만일 아무것도 적혀져 있지 않은 입력란을 제출하려고 한다면 단순 리턴 시키고 제출을 완료하면 입력창을 비워 다른 폼을 입력 받아 제출할 수 있도록 합니다.