2024-04-03 5.7.9 Hook - 2

현행 문서: 5.7.9 Hook – 2

1 React hook 실습

react Hook을 사용해 보세요!

Component

```
import React, {ChangeEvent, FC, useState} from "react";

const Hook2: FC = () => {
    const [age, setAge] = useState(11);
    const [genderCode, setGenderCode] = useState('M');

    const handleAgeChange = ({target: {value}}: ChangeEvent<HTMLInputElement>) => setAge(Number(value));
    const handleGenderCodeChange = ({target: {value}}: ChangeEvent<HTMLInputElement>) => setGenderCode(value);

    return (
        <>
            <User age={age} genderCode={genderCode}/>
            <input type="number" value={age} aria-label="age" onChange={handleAgeChange}/>
            <div>
                <label>
                    <input type="radio" value="M" name="genderCode" checked={genderCode === 'M'} onChange=
{handleGenderCodeChange}/>
                    M
                </label>
                <label>
                    <input type="radio" value="F" name="genderCode" checked={genderCode === 'F'} onChange=
{handleGenderCodeChange}/>
                    F
                </label>
            </div>
        </>
    )
}

export default Hook2;

interface UserProps {
    age: number;
    genderCode: string;
}

const User: FC<UserProps> = ({age, genderCode}) => {
    const generation = getGeneration(age);
    const gender = getGender(genderCode);

    return (
        <>
            <div>{generation}</div>
            <div>{gender}</div>
        </>
    )
}

const getGeneration = (age: number) => {

    console.log('Run getGeneration');

    if (age < 10) return '';
    return `${(age / 10) * 10}`;
}

const getGender = (genderCode: string) => {

    console.log('Run getGender');

    return genderCode === 'M' ? '' : '';
};
```

Component Test

```
import React, {render, screen} from '@testing-library/react';
import userEvent from "@testing-library/user-event";
import Hook2 from "./Hook2";

describe('Hooks Test', () => {
    /*
    * age  genderCode   getGeneration, getGender   . (  )
    *  Hook    .
    * */

    const log = jest.spyOn(global.console, "log").mockImplementation();
    const renderHook = () => {
        render(<Hook2/>);

        log.mockReset(); //
    }
    const changeAge = (age: number) => userEvent.type(screen.getByLabelText('age'), age.toString());
    const changeGenderCode = (genderCode: string) => userEvent.click(screen.getByLabelText(genderCode));

    afterEach(() => {
        log.mockReset();
    });

    test('age  Run getGeneration .', () => {
        renderHook();

        changeAge(33);

        expect(log).toBeCalledWith('Run getGeneration');
    });

    test('age  Run getGender  .', () => {
        renderHook();

        changeAge(24);

        expect(log).not.toBeCalledWith('Run getGender');
    });

    test('genderCode   Run getGender  .', () => {
        renderHook();

        changeGenderCode('F');

        expect(log).toBeCalledWith('Run getGender');
    });

    test('genderCode   Run getGeneration   .', () => {
        renderHook();

        changeGenderCode('F');

        expect(log).not.toBeCalledWith('Run getGeneration');
    });
})
```