

리팩터링

■ 개념

- 외부 동작(기능)을 바꾸지 않으면서 내부 구조를 개선하는 방법입니다.
- 코드가 작성(기능완성)된 후에 디자인을 개선하는 작업입니다.
- 모든 것을 미리 생각하기보다는 개발을 하면서 지속적으로 좋은 디자인을 찾는다.

■ 시작

뭔가 이상한 부분을 찾아보세요.

```
var numArr = [ 1, 2, 3, 4, 5 ];

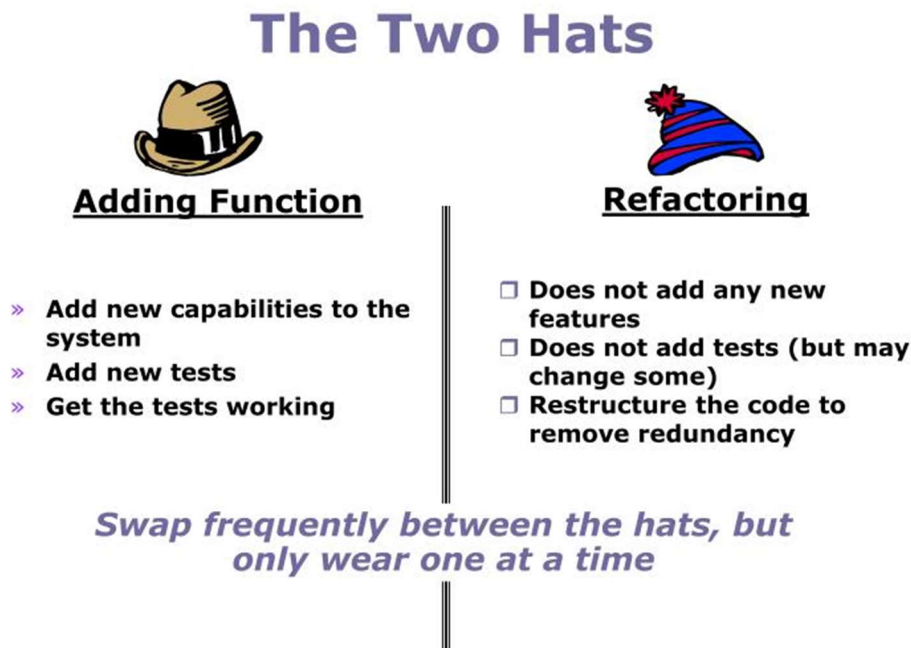
var length=0;
var i = 0;
while(true){
    if( numArr[i] != undefined ){
        i++;
        length++;
    }else{
        break;
    };
};

for( i=0 ; i<length ; i++){
    console.log(numArr[i])
};
```

■ 리팩터링 단계

- 리팩터링 모자 쓰기.

기능을 추가할 때는 기존 코드를 건드려서는 안 되고 단지 새로운 기능만 추가해야 한다
리팩터링을 할 때는 기능을 추가해서는 안 되고 단지 코드의 구조변경에만 신경써야 한다.



- 리팩터링은 작은 단계로 나누어 프로그램을 변경하고
원래대로 작동하는지 바로 실행해서 확인한다. (단위 테스트를 작성해서 확인하면 편리하다.)

■ 리팩터링 방법

- 목적이 잘 드러나게 Naming하기.
컴퓨터가 이해하는 코드는 누구나 작성할 수 있다.
좋은 프로그래머는 사람이 이해할 수 있는 코드를 작성한다.
- 관심사의 분리
관심사의 분리는 서랍에 비슷한 물건들끼리 잘 정리해서 넣어두는 것과 같다.
서랍에 잘 정리해두면 필요할 때 물건을 빠르고 편하게 찾을 수 있다.

다음 코드는 몇개의 관심사를 가지고 있을 까요?

```
function gridRowSum(){
    var datainformation = $('#jqxgrid').jqxGrid('getdatainformation');
    var rowcount = datainformation.rowcount;
```

```

var sum = 0;
for (var m = 0; m < rowcount; m++) {
    var rowData = $("#jqxgrid").jqxGrid('getrowdata', m);
    sum += rowData.PAY;
};
return sum;
};

```

관심사는 크게 3가지 정도로 볼수 있다.

1. grid의 row count를 가져오는 관심사.
2. grid의 row 값을 가져오는 관심사.
3. grid의 row 값을 합하는 관심사.

여기에 남아 있어야 할 관심사 1개와 불필요한 관심사 2개는 무엇일까요?

여기서 중심이 되는 로직은 반복문이고
반복문은 grid의 row 값을 합하기위해 존재합니다.
그러므로 남아있어야 할 관심사는 row값을 합하는 것이다.
이외의 관심사는 불필요 합니다.

- 함수를 사용해서 관심사 분리하기.
- 임시변수를 함수로 추출하기.

```

function gridRowSum(){
    var datainformation = $('#jqxgrid').jqxGrid('getdatainformation');
    var rowcount = datainformation.rowcount;

    var sum = 0;
    for (var m = 0; m < rowcount; m++) {
        var rowData = $("#jqxgrid").jqxGrid('getrowdata', m);
        sum += rowData.PAY;
    };
    return sum;
}

```

```
};
```

```
function gridRowSum(){
    var sum = 0;
    for ( var i = 0 ; i < rowsCount() ; i++ ) {
        sum += rowDataPay(i);
    };
    return sum;
};

function rowsCount(){
    var datainformation = $('#jqxgrid').jqxGrid('getdatainformation');
    return datainformation.rowscount;
};

function rowDataPay( index ){
    var rowData = $('#jqxgrid').jqxGrid('getrowdata', index );
    return rowData.PAY;
};
```

- 중첩부분 함수로 추출하기.
- 클래스로 추출하기.

■ 참조

<https://nesoy.github.io/articles/2018-05/Refactoring>