

프로젝트 명세서

Collection 사용

목차

1. 개요	3
2. 예시 #1.....	4
3. 예시 #2.....	5
4. 예시 #3.....	6
5. Collection 장단점	8
6. 과제	9

1. 개요

코드에서 비교 연산 후 결정 되는 동작이, 동일한 패턴으로 반복 된다면 Collection 을 사용 하는 방법이 있습니다. 이 방법이 적용 가능한 경우를 자주 만나게 되고, 적용 했을 경우, 가독성, 성능 등이 개선 될 수 있습니다. 이 방법은 특정 랭귀지에 국한되지 않고, 모든 랭귀지에서 사용 가능 합니다.

여기서 말하는 Collection 이란, Stack, Queue, Array, List, Hash, Map, Dictionaty 같은 데이터를 그룹으로 저장 할 수 있는 메모리 구조를 말합니다.

2. 예시 #1

아래 Java 코드는 랜덤으로 과일 이름을 리턴 하는 함수 입니다. switch block 내부를 보시면 case 0, 1, 2, 3 에 따라 과일명만 다를 뿐, 리턴 한다는 동작 자체는 동일하게 반복 되네요.

```
public String getFruit(){  
  
    int random = RandomInt();  
  
    int i = random % 4;  
  
    switch(i) {  
        case 0:  
            return "Mango";  
        case 1:  
            return "Apple";  
        case 2:  
            return "Orange";  
        case 3:  
            return "Banana";  
    }  
}
```

이런 경우, 아래와 같이 array 에 과일 이름을 저장 후, index 로 접근하여 과일 이름을 리턴 하는 방법도 있습니다.

```
public string getFruit(){  
  
    String fruits[] = {"Mango", "Apple", "Orange", "Banana"};  
  
    int random = RandomInt();  
  
    int i = random % fruits.getLength();  
  
    return fruits[i];  
}
```

3. 예시 #2

아래 java-script 코드는 사용자가 게시판에 작성한 글을 저장 기능으로, frontend 에 해당하는 함수 입니다. user, title, body, image 변수들의 값이 있는 검사하여, 변수에 값이 없을 경우 그에 해당하는 alert 을 띄웁니다.

```
function post() {

    if(this.user == '') {
        alert('로그인 해주세요');
        return;
    }
    else if(this.title == '') {
        alert('제목을 입력해 주세요');
        return;
    }
    else if(this.body == '') {
        alert('내용을 입력해 주세요');
        return;
    }
    else if(this.image == '') {
        alert('이미지를 첨부하세요');
        return;
    }

    return postCore(this.user, this.title, this.body, this.image)
}
```

이런 경우, 아래와 같이 array 구조를 만들어서, for 문으로 처리하는 방법도 있습니다.

```
function post() {

    const checkList = [
        [this.user, '로그인 해주세요'],
        [this.title, '제목을 입력해 주세요'],
        [this.body, '내용을 입력해 주세요'],
        [this.image, '이미지를 첨부하세요']]

    for(const v of checkList) {
        if(v[0] == '') {
            alert(v[1]);
            return;
        }
    }

    return postCore(this.user, this.title, this.body, this.image)
}
```

4. 예시 #3

아래 python 코드는 비행기 게임을 할 때, 입력 키 값에 따라 방향 함수가 호출되는 코드입니다.

```
def forward():  
    print('move forward')  
  
def left():  
    print('move left')  
  
def right():  
    print('move right')  
  
def backward():  
    print('move backward')  
  
def move(key):  
  
    if(key == 'w'):  
        forward()  
    elif(key == 'a'):  
        left()  
    elif(key == 'd'):  
        right()  
    elif(key == 's'):  
        backward()
```

이런 경우, 입력 키와 그에 해당하는 함수를 Mapping 하는 dictionary 구조를 만들어서 처리 하는 방법도 있습니다.

```
def forward():  
    print('move forward')  
  
def left():  
    print('move left')  
  
def right():  
    print('move right')  
  
def backward():  
    print('move backward')  
  
def move(key):  
  
    dict = {'w':forward, 'a':left, 'd':right, 's':backward}  
  
    func = dict.get(key)  
  
    if func is not None:  
        func()
```

5. Collection 장단점

```

public class A {

    public String getFruit() {

        int random = RandomInt();

        int i = random % 4;

        switch(i) {

            case 0:
                return "Mango";
            case 1:
                return "Apple";
            case 2:
                return "Orange";
            case 3:
                return "Banana";
            default:
                return null;
        }
    }
}

```

```

public class B {

    String fruits[] = {"Mango", "Apple", "Orange", "Banana"};

    public String getFruit() {

        int random = RandomInt();

        int i = random % this.fruits.length;

        return this.fruits[i];
    }
}

```

예제#1을 위와 같이 A와 B를 Class로 만들었습니다. 두 클래스는 동일한 기능을 하지만 다음과 같은 차이점 있습니다.

- 힙 메모리 사용량 => A < B
- 함수(getFruit) 실행 속도 => A < B

두 방식 모두 장/단점이 있기 때문에 어느 방식을 적용 할 지는, 요구사항, 메모리 사용량, 가독성, CPU 연산량 등 다각도로 고려 해야 합니다.

6. 과제

```
const date = new Date()
const year = date.getFullYear()
const month = date.getMonth()

var days = null

switch(month){

    case 0:
    case 2:
    case 4:
    case 6:
    case 7:
    case 9:
    case 11:
        days = 31;
        break;

    case 3:
    case 5:
    case 8:
    case 10:
        days = 30;
        break;
    case 1:
        if((year % 4 == 0) && (year % 100 != 0) || (year % 400) == 0)
            days = 29;
        else
            days = 28;
        break;
}

console.log(days + ' days for ' + year + '-' + (month + 1))
```

위의 코드는 현재 달의 날수를 출력 합니다. 특히 2 월 달은 윤년, 평년에 따라 날 수가 결정 됩니다.

과제 : switch 대신, collection 을 사용하는 방법으로 코드를 작성해서 제출 하세요.

- Language 는 C++, Java, java-script, python 등 무관하게 선택 가능

- 정답은 복수 개가 될 수 있고, 한 개 이상 제출 해도 무방, 단, 복수 제출시 가산 점 없음
- 이미 제공하는 API 를 사용하지 말고, 반드시 collection 을 사용하여 개발 할 것.

**** 아래와 같이 이미 제공하는 API 사용하지 말 것 ****

<https://stackoverflow.com/questions/1184334/get-number-days-in-a-specified-month-using-javascript/1464716>

<https://stackoverflow.com/questions/8821457/get-the-number-of-days-in-the-current-month-in-java>

7. 심화과제

공통 프로젝트에서 Collection 을 사용해서 처리 할 수 있는 부분이 있다면, 찾아서
고치시오.