문제은행 [2] 복습풀이+ 질문

12/20 개강 SW개발자 양성과정 [손현지]

2021. 12. 26

〈문제은행[2] - 1번, Answer1 보며 다시 풀이하기〉

```java
//대문자 출력을 위한 식
boolean condition1 = random >= 65 && random <= 90;
//풀이: int random이 65보다 크거나 같고 또한 int random이 90보다 작거나 같으면 참이다.

//소문자 출력을 위한 식
boolean condition2 = random >= 97 && random <=122;
//풀이: int random이 97보다 크거나 같고 또한 int random이 122보다 작거나 같으면 참이다.

if(condition1){ //만약, condition1의 조건을 만족하는 경우 아래의 내용을 출력한다.
    System.out.printf("65~90사이의 랜덤한 문자 생성(대문자) : %c\n", random);
}

if(condition2){ //만약, condition2의 조건을 만족하는 경우 아래의 내용을 출력한다.
    System.out.printf("97~122사이의 랜덤한 문자 생성(소문자) : %c\n", random);
}
```

기존의 코드

-문제를 잘못 이해하여
영어대소문자 외의
기호(아스키코드 91~96번)이
출력되지 않았음.

해결을 위한 전략 세우기

*반복문 두 개가 사용되어야 함

반복문

〈아스키코드 A ~ z(65~122)까지〉

여기로 돌아감

그 중에서 대문자/소문자만
출력해주는 반복문

*break를 걸어주면 여기에서 출력 종료

문자가 아닌 기호(91~96)을 출력하는 printf

```java
public class Answer1 {
    public static void main(String[] args) {
        //문제1. 65 ~ 122 사이의 랜덤한 문자를 생성하도록 한다.
        //      여기서 소문자나 대문자가 아니라면 다시 생성하도록 프로그램을 만들어보자.
        // 복습 포인트: fianl int를 사용해보기! ---------------------------------
        //           아스키코드 기호가 나오면 한줄이 더 출력되도록 if else 사용하여 제작하기.

        final int MAX = 122;//출력되는 가장 작은 숫자
                            //또한 65는 대문자 A의 10진수 아스키 코드 번호! 대문자 Z는 90
        final int MIN = 65; //출력되는 가장 큰 숫자
                            //97~122는 소문자 a~z 의 아스키 코드 번호!

        int range = MAX - MIN +1; // range = 122-65 = 57(min과 max 두 숫자 사이의 범위)
                                  // 하지만 그냥 57은 0~56으로 출력되니까 +1을 더한다!
        boolean isChar = true; //이것도 미리 설정해두면 나중에 isChar 의 값만 바꿔주면 되니 편하다.

        while(isChar){ //while의 조건문이 true이면 반복문이 무한반복
            int rand = (int)(Math.random() * range + MIN);
                // range(1~57의 숫자)에 모두 +65가 더해져 122까지의 숫자가 출력될 수 있음.
                // 또한 반복문 안에 random() 메서드가 있어야 문장 다시 출력하기가 가능!

            boolean condition1 = rand >= 65 && rand <= 90;
            boolean condition2 = rand >= 97 && rand <= 122;

            if(condition1 || condition2){ // ||의 의미는 or 이다. 좌항 우항 중 하나라도 참이면 전체가 참이 되어 값이 출력.
                System.out.printf("rand는 영어 대소문자 %c(%d)\n", rand, rand);
                isChar = false;
                break;
            }

            System.out.printf("rand는 문자가 아닌 기호 %c(%d) 영어 대소문자가 나올 때 까지 재출력\n", rand, rand);
        }
    }
}
```

반복문1

반복문2

랜덤값부터 다시 구하러 GO

출력결과

```
rand는 문자가 아닌 기호 `(96) 영어 대소문자가 나올 때 까지 재출력
rand는 영어 대소문자 H(72)
```

```
rand는 영어 대소문자 o(111)
```

질문사항

1. 다른 부분은 다 이해했는데 여기에서 isChar가 왜 들어가는지 모르겠습니다.

1-1.또, 결과값을 false로 바꿔줘야 하는 이유를 모르겠습니다. 어차피 break만 있으면 반복문 while을 빠져나갈 수 있는 거 아닌가요?

2. 값을 true에서 false로 바꿔주는 거라면 !(not)을 사용하여 코드를 입력할 수도 있는지 궁금합니다.

〈문제은행[2] – 2번, Answer2 보며 다시 풀이하기〉

기존의 코드

```java
//문제2. 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ... 일명 피보나치 수열의 20번째 항을 구하도록 프로그램 해보자!
//피보나치 수열이란?
        // 첫째 및 둘째 항이 1이며 그 뒤의 모든 항은 바로 앞 두 항의 합인 수열이다.

int i = 1;
int j = 1;
int sum = 0;

int again = 1;

System.out.printf("%d, %d, ", i, j); //맨 앞의 1, 1 한 번만 나옴
while( again <= 18){ //맨 앞 두 항(1, 1)를 포함한 20번째 항이 나올 때 까지
    again++; // 반복마다 +1해가며 18회를 돈다.

                // int i + int j = sum의 반복되도록 만드려고 함!
    sum = i + j; // sum은 i + j 값이다.

    System.out.printf("%d, \n", sum); //sum의 값을 출력

    // 매회 i > j, j > sum, sum > 다음 합 으로 변수를 변경해주어야 함!
    i = j;
    j = sum;

}

System.out.println("피보나치 수열 20번째 항의 값은 = " + sum);
```

```java
// 복습 포인트: fianl int를 사용해보기! ---------------------------------

int num1 = 1;
int num2 = 1;
int sum = 0, i; → sum = i = 0 으로 값이 모두 같음

final int start = 2;
final int end = 20;

System.out.printf("%d, %d, ", num1, num2);

for (i = start; i < end; i++) {
    // i값을 2로 바꿔줌; i의 값이 20이 될 때 까지; 매 회 +1씩 숫자가 커지며 반복한다는 조건을 만족하는 경우

    sum = num1 + num2; → sum(2)의 값을 num1(1) + num2(1) 로 변경

    System.out.printf("%d, ", sum);

    num1 = num2;
    num2 = sum;

}

System.out.printf("\n피보나치 수열 %d번째 항의 값은 = %d", i, sum);
```

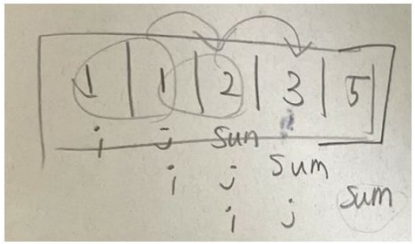
i = start(2)로 값을 바꿔주면서 sum = i = 2 로 값이 변경

출력결과

```
1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765,
피보나치 수열 20번째 항의 값은 = 6765
```

해결을 위한 전략 세우기

*(12/23 목요일 – 구조를 그려 이해해보기)



질문사항

1. 컴퓨터는 배열의 시작이 0이기 때문에 반복 시작 기준을 0으로 맞춰주는 것이 좋다고 적어두셨어요. 이 부분을 i를 가장 처음에 0으로 맞춰두신 건 알겠는데... 효율적인 건지는 아직 잘 이해가 가지 않아요. 아직 코드가 짧은 편이라 그런걸까요? 아니면 관습적인 부분인가요?

2. final int 변수명은 왜 조건문에 넣을 수가 없나요?

3. int의 이름은 관습적으로 소문자를 쓰는 편인데 왜

4. 이후 반복문 조건 속 i가 sum 값과 상관없이 +1씩 증가하는 건 이미 조건은 먼저 실행되었기 때문에 이후 적용된 식의 영향을 받지 않기 때문인가요? i의 값이 sum과 같은데 마지막 printf에서는 i값과 sum 값이 다르게 나오게 되어서 궁금해졌습니다.

〈문제은행[2] – 4~8번, Answer 보며 다시 풀이하기〉

기존의 코드

```java
int sum = 0; //우선 sum 메모리에 들어가는 숫자를 0으로 설정한다.

for (int i = 1; i <= 100; i++){
                    //i는 1이다; i의 숫자가 100보다 작거나 같을 때 까지 반복; 줄마다 i의 수가 +1씩 커짐
    if(i % 4 == 0){ //만약 ' 4로 나누면 0이 됩니까?'라는 조건문을 만족한 i 숫자들의 경우
        sum += i;   //1~100내에서 마지막 4의 배수에 도달할 때 까지 더해진다.
    }
}

System.out.println("1~100중 4의 배수인 숫자들을 모두 더한 결과는 = " + sum);
```

```java
//문제8. 1 ~ 100까지 숫자를 순회한다
    // 이 부분 조건이 있는 무한루프 만들라는 게 아니라 1~100 한바퀴 나오게 하라는 말씀이 맞죠??
//2 ~ 10 사이의 랜덤한 숫자를 선택하고 이 숫자의 배수를 출력해보도록 한다.

int i = 0;
int random = (int)(Math.random() *9 + 2);

System.out.println("random 의 값은 = " + random);

for(i =1; i<=100; i++){
    if(i % random == 0){ // i를 random난숫의 숫자로 나누면 남는 값이 0이 된다면
        System.out.printf("1~100 사이에서 %d의 배수 : %d\n", random, i); //이 내용을 출력
                            //앞의 %d는 random의 수 대입, 뒤의 %d는 i의 수를 대입

    }
}
```

final int를 사용하지 않음

사실 복습 도중 두 번이나 발생한 오류의 원인을 모르겠어서
질문하려고 이 페이지를 만든 거였는데... 다시 입력해보니 오류가 안 뜨네요...
뭐였을까요... 그래도 그냥 복습한 내용은 남겨둡니다.

```java
for (int i = start; i <= end; i++) {
        if (i % div == remain) {
        System.out.printf("%d의 배수 = %d\n", div, div * i);
        }
```

이런 식의 반복문을 쳤을 때 동그라미 친 부분의 i가 빨간색으로 변하면서 i를 찾을 수 없다는 안내가 나왔었습니다.
(캡쳐상에서 전체적으로 빨간 밑줄 있는 건 상관x입니다. class 바깥쪽에 복붙해두고 캡쳐해서 그런 거예요!)

복습 결과

```java
//문제4~8. 배수만들기 문제 + 배수할 수를 랜덤으로 정하기 + 배수들을 더한 결과 출력시키기
// 복습 포인트: fianl int를 사용해보기! ---------------------------------

int div = (int) (Math.random() * 10 + 1); // 구할 배수 값을 랜덤으로
final int start = 1;
final int end = 100;
final int remain = 0;

int sum = 0;

for (int i = start; i <= end; i++) {
    if (i % div == remain) {
        System.out.printf("%d의 배수 = %d\n", div, i);

        sum += i;
    }
}

System.out.printf("%d의 배수를 모두 구한 값 = %d", div, sum);
```

출력결과

```
9의 배수 = 9
9의 배수 = 18
9의 배수 = 27
9의 배수 = 36
9의 배수 = 45
9의 배수 = 54
9의 배수 = 63
9의 배수 = 72
9의 배수 = 81
9의 배수 = 90
9의 배수 = 99
9의 배수를 모두 구한 값 = 594
```

```
6의 배수 = 6
6의 배수 = 12
6의 배수 = 18
6의 배수 = 24
6의 배수 = 30
6의 배수 = 36
6의 배수 = 42
6의 배수 = 48
6의 배수 = 54
6의 배수 = 60
6의 배수 = 66
6의 배수 = 72
6의 배수 = 78
6의 배수 = 84
6의 배수 = 90
6의 배수 = 96
6의 배수를 모두 구한 값 = 816
```

```
1의 배수 = 87
1의 배수 = 88
1의 배수 = 89
1의 배수 = 90
1의 배수 = 91
1의 배수 = 92
1의 배수 = 93
1의 배수 = 94
1의 배수 = 95
1의 배수 = 96
1의 배수 = 97
1의 배수 = 98
1의 배수 = 99
1의 배수 = 100
1의 배수를 모두 구한 값 = 5050
```

〈문제은행[2] – 9번, Answer9 보며 다시 풀이하기(1)〉

기존의 코드

```java
int sum = 0; //우선 sum메모리에 들어가는 숫자를 0으로 설정한다.

for(int i = 1; i <=100; i++){ // 1~100까지 숫자가 나올 때 까지만 반복

    int random = (int)(Math.random() *9 +2);
        // for문 밖에 있을 땐 코드 전체에서 한 번 밖에 실행되지 않기 때문에 결과가 1개였지만
        // for문 안에 넣으니 반복되는 동안 계속 새로운 숫자로 출력됨!

    if(i % random == 0){ //'int 랜덤숫자'의 배수만 출력할 수 있는 조건문
        System.out.println("랜덤 숫자의 값 = " + random);
        System.out.printf("랜덤 숫자의 배수 i의 값 = %d\n", i);

        sum += i; // if문이 100회 반복되는 동안 숫자가 계속 더해진다.
    }
}

System.out.println("랜덤 숫자의 배수 i의 값을 모두 더한 결과는 = " + sum ); //100회 반복되며 더해진 결과 출력
```

출력결과

(전체출력 결과가 길어서 위쪽을 잘랐습니다)

```
랜덤 숫자의 값 = 3
랜덤 숫자의 배수 i의 값 = 81
랜덤 숫자의 값 = 4
랜덤 숫자의 배수 i의 값 = 84
랜덤 숫자의 값 = 2
랜덤 숫자의 배수 i의 값 = 86
랜덤 숫자의 값 = 2
랜덤 숫자의 배수 i의 값 = 90
랜덤 숫자의 값 = 3
랜덤 숫자의 배수 i의 값 = 93
랜덤 숫자의 값 = 9
랜덤 숫자의 배수 i의 값 = 99
랜덤 숫자의 배수 i의 값을 모두 더한 결과는 = 1048
```

```
랜덤 숫자의 값 = 2
랜덤 숫자의 배수 i의 값 = 76
랜덤 숫자의 값 = 8
랜덤 숫자의 배수 i의 값 = 80
랜덤 숫자의 값 = 5
랜덤 숫자의 배수 i의 값 = 85
랜덤 숫자의 값 = 4
랜덤 숫자의 배수 i의 값 = 88
랜덤 숫자의 값 = 8
랜덤 숫자의 배수 i의 값 = 96
랜덤 숫자의 값 = 5
랜덤 숫자의 배수 i의 값 = 100
랜덤 숫자의 배수 i의 값을 모두 더한 결과는 = 854
```

- 정수 숫자를 직접 입력하는 대신 final int를 사용하여 유지보수가 편하도록 변경
- boolean을 사용하여 무한반복의 방어(?)를 만들어줌
- 결과물은 똑같이 나올 수 있도록 함

해결을 위한 전략 세우기

*(12/23 목요일 - 구조를 그려 이해해보기)

반복문

1.Math.random()
- 반복문 안에 넣어서 루프마다 새 숫자가 나오도록 하기

반복문2

2. if(i를 랜덤 수로 나누면 남는 값이 0일 때)
- { 랜덤숫자의 값 출력, 랜덤숫자의 배수 i의 값 출력,
  boolean = false;,
  sum + i로 배수의 값들 더해주기 }

복습 결과

```java
//문제9. 1 ~ 100까지 숫자를 순회한다.
//2 ~ 10 사이의 랜덤한 숫자를 선택하고 이 숫자의 배수를 출력한다.
//다음 루프에서 다시 랜덤 숫자를 선택하고 해당 숫자의 배수를 출력한다.
//끝까지 순회했을 때 출력된 숫자들의 합은 얼마인가?
// 복습 포인트: final int 사용, boolean을 이용하여 방어??? 시키기. ----------------------------

final int start = 1;
final int end = 100;
final int remain = 0;

boolean defense = true;
int sum = 0; //우선 sum메모리에 들어가는 숫자를 0으로 설정한다.

for(int i = start; i <= end; i++){ // 1~100까지 숫자가 나올 때 까지만 반복

    int random = (int)(Math.random() *9 +2);
    // for문 밖에 있을 땐 코드 전체에서 한 번 밖에 실행되지 않기 때문에 결과가 1개였지만
    // for문 안에 넣으니 반복되는 동안 계속 새로운 숫자로 출력됨!

    if(i % random == remain){ //'int 랜덤숫자'의 배수만 출력할 수 있는 조건문
        System.out.println("랜덤 숫자의 값 = " + random);
        System.out.printf("랜덤 숫자의 배수 i의 값 = %d\n", i);
        defense = false;

        sum += i; // if문이 100회 반복되는 동안 숫자가 계속 더해진다.
    }
}

System.out.println("랜덤 숫자의 배수 i의 값을 모두 더한 결과는 = " + sum ); //100회 반복되며 더해진 결과 출력
```

〈문제은행[2] - 9번, Answer9 보며 다시 풀이하기(2)〉

선생님 답 - Answer 9

```
1   public class Answer9 {
2       public static void main(String[] args) {
3           final int START = 1;
4           final int END = 100;
5           final int REMAIN = 0;
6
7           final int MAX = 10;
8           final int MIN = 2;
9
10          int range = MAX - MIN + 1;
11          /*  1 ~ 100까지의 숫자를 순회한다.
12              2 ~ 10 사이의 랜덤한 숫자를 선택하고 이 숫자의 배수를 출력한다.
13              다음 루프에서 다시 랜덤 숫자를 선택하고 해당 숫자의 배수를 출력한다. */
14
15          // 구현 전략이 필요함
16          // 1. 1 ~ 100까지 숫자 순회는 for문을 사용
17          // 2. for 문 내부에서 랜덤 난수 생성 2 ~ 10
18          // 3. 난수의 배수를 검사하기 전까지 랜덤을 다시 생성하면 안됨
19
20          // 실제 난수의 할당(생성) 했는지 안했는지 판정 여부
21          boolean isRandomAllocCheck = false;
22          int decision = 0;
23  +       int sum = 0;
24
25          for (int i = START; i <= END; i++) {
26              while (!isRandomAllocCheck) {
27                  decision = (int) (Math.random() * range + MIN);
28                  isRandomAllocCheck = true;
29              }
30
31              if (i % decision == REMAIN) {
32                  System.out.printf("%d의 배수 i = %d\n", decision, i);
33                  isRandomAllocCheck = false;
34  +
35  +               sum += i;
36              }
37          }
38  +
39  +       System.out.println("현재까지 나타난 숫자들의 합 = " + sum);
40      }
41  }
```

반복문1

반복문2

난수가 할당되었다면 반복문 3으로?

???

반복문3

질문사항

1-1. Answer1을 다시 풀면서 남긴 질문과 이어지는 내용이 될 것 같아요.
선생님께서 이걸 보실 때 쯤엔 제가 어느 정도 이해를 해야 할텐데...

제가 푼 방식에서는 boolean 변수가 있어도 없어도 딱히 달라지는 점이 없어서 왜 boolean 변수 = false;(방어?)를 넣어줄 필요가 있는 건지 역시 모르겠어요.

1-2. 선생님께서 답을 만드실 때 Answer1 도 Answer9도 while(true) 반복문을 사용하셨더라구요.
혹시 boolean 변수 = false;(방어) 는 while로 무한반복 반복문을 만들었을 때만 필요한 것은 아닌가요?
만약 그렇다면 제가 처음 숙제로 만들었던 기존 내용은 while반복문을 사용하지 않았고, for(조건)을 만족하면 반복이 멈추는데 왜 반복문을 멈추기 위한 boolean이 별도로 필요한 걸까요?

→ 난수를 할당하지 않는 경우도 있는 건가?(오류가 난다거나??)
그런 경우에도 랜덤을 반복하면 문제가 생길 수 있고,
불필요하게 랜덤이 돌아가기 때문에 효율이 떨어져서 판정해야 하는건가?
→ 이것 때문에 반복문 3개로 만드신 건가?

→ 이 부분이 난수가 할당됐는지 검사하는 부분?

이걸 적으면서 느끼는건데 역시 제가 boolean으로 방어를 한다는 말씀이 무슨 뜻이었는지 아직 잘 이해를 못한 것 같습니다. 혹시 boolean 변수 = false;가 있을 때와 없을 때 결과가 달라지는 예제가 있을까요?ㅠ 눈으로 보지 않으면 이해가 잘 되지 않을 것 같아서요...

이걸 다시 뜯어보니 복습을 하면서도 Math.random()에 final int써보는 건 또 까먹었네요.
commit - push 하기 전에 고쳐서 올릴게요.