

컴프2 1주차 : 별찍기, 버블정렬의 구현

과제 목표 : 별찍기와 버블정렬을 구현하여 조건문과 반복문을 활용할 수 있음을 보인다.

학번 : 201902722

이름 : 유형곤

1번 과제 : 별찍기

문제해결과정 및 코드 설명 :

우선 1번 문제는 크게 3가지의 문제로 나눌 수 있습니다.

1. 삼각형 모양의 별 그리기

삼각형 모양은 비교적으로 단순합니다. 우선 규칙성을 살펴보면, 삼각형의 높이를 h 라고하면,

총 h 개의 라인을 출력하는데, 라인이 늘어날 때마다 별을 한 개씩

이를 표현하면 아래와 같습니다.

* (별이 1개)

** (별이 2개)

.... (...중략...)

***** (별이 h 개)

코드는 다음과 같습니다.

```
//삼각형
System.out.println("input line numbers:");
h = input.nextInt();
System.out.println(String.format("---[%d]---", tries++));
for(int i = 1; i <= h; i++) {
    for(int j = 0; j < i; j++) {
        bw.write("*");
    }
    bw.write("\n");
}
bw.flush();
```

(속도 개선을 위하여 버퍼에 문자열을 먼저 기록하고 한꺼번에 출력했습니다.)

2. 역삼각형 모양의 별 그리기

역삼각형은 규칙이 더 많습니다. 우선 라인 수는 h 로 1번과 같지만, 라인이 늘어날 때마다 별의 개수가 줄어들고, 공백의 개수가 늘어납니다.

이를 표현하면 다음과 같습니다.

```
***** (별이 h개, 공백이 0개)

**** (별이 h-1개, 공백이 1개)

***

**

* (별이 1개, 공백이 h-1개)
```

이를 코드로 표현하면 다음과 같습니다.

```
//역삼각형
System.out.println("input line numbers:");
h = input.nextInt();
System.out.println(String.format("---[%d]---", tries++));
for(int i = 0; i < h; i++) {
    for(int j = 0; j < i; j++) {
        bw.write(" ");
    }
    for(int j = h - 1 - i; j >= 0; j--) {
        bw.write("*");
    }
    bw.write("\n");
}
bw.flush();
```

3. 다이아몬드 모양의 별 그리기

다이아몬드 같은 경우는 (int) $h/2 + 1$ 개의 라인과 나머지 (int) $h/2$ 개의 라인의 규칙이 다르기 때문에 두 개의 반복문으로 나누어 구현했습니다.

$h = 5$ 라고 할 때, 규칙은 다음과 같습니다.

```
* (별이 1개, 공백이 h/2+1-1개)

*** (별이 3개, 공백이 h/2+1-2개)

***** (별이 5개, 공백이 h/2+1-3개)
```

여기까지는 별이 2개씩 증가하고, 공백이 2개씩 감소합니다.

*** (별이 3개, 공백이 1개)

* (별이 1개, 공백이 2개)

여기부터는 공백이 1개씩 증가하고, 별이 2개씩 감소합니다.

이를 코드로 구현하면 다음과 같습니다.

```
//다이아몬드
System.out.println("input line numbers:");
h = input.nextInt();
System.out.println(String.format("---[%d]---", tries++));
for(int i = 1; i <= h / 2 + 1; i++) {
    for(int j = 0; j <= h / 2 - i; j++) {
        bw.write(" ");
    }
    for(int j = 0; j < 2*i-1; j++) {
        bw.write("*");
    }
    bw.write("\n");
}
for(int i = 1; i <= h / 2; i++) {
    for(int j = 0; j < i; j++) {
        bw.write(" ");
    }
    for(int j = 0; j < h-2*i; j++) {
        bw.write("*");
    }
    bw.write("\n");
}
bw.flush();
```

실행결과 : (사진은 다음 페이지에 있습니다.)

```
osboxes@osboxes: ~/Desktop/COMP2/CP2_201902722_01
[INFO] -----[ jar ]-----
[INFO] --- exec-maven-plugin:1.6.0:java (default-cli) @ CP2_201902722_01 ---
input line numbers:
7
---[1]---
*
**
***
****
*****
*****
*****
input line numbers:
7
---[2]---
*****
*****
****
***
**
*
input line numbers:
7
---[3]---
*
***
*****
*****
*****
***
*
osboxes@osboxes:~/Desktop/COMP2/CP2_201902722_01$
```

2번 과제 : 버블정렬 구현

문제해결과정 및 코드 설명 :

1번 문제 같은 경우는 정해지지 않은 개수의 숫자를 입력받아야 합니다. 따라서 저는 우선 한 라인을(문자열로) 통째로 읽고, 이를 StringTokenizer를 이용하여 숫자를 하나씩 가져와서 배열에 저장했습니다. 그런 후 일반적인 버블정렬 알고리즘을 구현했습니다. 버블정렬 알고리즘은 두 원소를 비교하여 정렬이 되어있지 않으면 교환해나가는 알고리즘으로, 배열을 1번 순회할 때마다, (정렬되지 않은 원소들 중) 가장 큰 원소가 정렬된다는 특징이 있습니다. 따라서 배열을 N번 (배열의 길이)만큼 순회해야 합니다. 배열을 1번 순회하는 데에 필요한 시간 복잡도가 $O(N)$ 이고, 이를 N번 반복해야 하므로 버블정렬의 시간복잡도는 $O(N^2)$ 이 됩니다. 이를 코드로 구현하면 다음과 같습니다.

```

//버블정렬 (오름차순)
for(int i = 0; i < array.length; i++) {
    for(int j = 0; j < array.length-i-1; j++) {
        if(array[j] > array[j+1]) { //두 원소가 정렬되어있지 않다면
            //값을 교환
            int tmp = array[j];
            array[j] = array[j+1];
            array[j+1] = tmp;
        }
    }
}
}

```

수행 결과 :

```

osboxes@osboxes: ~/Desktop/COMP2/CP2_201902722_01
[INFO] -----
[INFO] Total time: 10.960 s
[INFO] Finished at: 2019-09-08T07:17:07-04:00
[INFO] -----
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by com.google.inject.internal.cglib.core.$ReflectUtils$1 (file:/usr/share/maven/lib/guice.jar) to method java.lang.ClassLoader.defineClass(java.lang.String,byte[],int,int,java.security.ProtectionDomain)
WARNING: Please consider reporting this to the maintainers of com.google.inject.internal.cglib.core.$ReflectUtils$1
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
[INFO] Scanning for projects...
[INFO] -----< com.github.hygoni:CP2_201902722_01 >-----
[INFO] Building CP2_201902722_01 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO] --- exec-maven-plugin:1.6.0:java (default-cli) @ CP2_201902722_01 ---
정렬 전 숫자 :
9 8 7 1 2 3 6 5 4 0
정렬된 결과:
0 1 2 3 4 5 6 7 8 9 osboxes@osboxes:~/Desktop/COMP2/CP2_201902722_01$

```