

# 프로그래밍 언어 활용

## 2조

서은빈 김준희 최서연 하태형

# 문제1 정삼각형 별짓기

```
Scanner sc = new Scanner(System.in);    //값을 입력받기 위해 클래스를 불러오는
                                         //객체 생성
System.out.printf("높이를 입력하세요 : "); // 높이 입력
int h = sc.nextInt();    //입력받는 높이 값
int i = 0; //높이(줄)의 기준점
while(i<h) //기준값인 i가 입력받은 h보다 작을 때를 범위로 잡고
           //증감호 안의 함수를 반복한다
{
    //공백 = n
    //별보다 공백이 먼저 입력
    int n=0;    //공백 초기 갯수
    while(n<(h-1)-i) //n이 직각 삼각형으로 배열
    {
        System.out.print(" "); //공백 입력
        n++;    //입력이 된 후 n값이 1씩 증가한다
    } //n의 값이 변할 때 여전히 조건에 부합할 시 입력 함수 반복
    //조건에 만족하지 않을 시 반복 탈출

    //별 = j
    //별은 공백 이후 배열
    int j = -1; //이하가 아닌 미만으로 잡았기 때문에 기준점이 -1이 된다
    while (j<2*i) { //범위식에서 오른쪽 값은 포함하지 않을
        System.out.print("*"); //별 입력
        j++;    //입력한 후 j값은 1씩 증가한다
    } //i값이 증가했을 때 반복문의 조건이 성립하는 값이면 반복문 실행
    //i값이 증가했을 때 반복문 조건을 만족하지 않으면 탈출

    System.out.println();

    i++;    // 위 반복문들이 모두 실행 종료되면 i값 1씩 증가
} //i값이 증가했을 때 반복문의 조건이 성립하면 반복문 실행
//i값이 증가했을 때 반복문의 조건을 만족하지 않으면 탈출
sc.close(); //Scanner 종료
```

## 풀이 결과 출력

```
높이를 입력하세요 : 7
      *
     ***
    *****
   *********
  ***********
 *****
*****
```

## 문제 풀이 과정

정삼각형(높이 받기) (입력값이 7일 때)

i = 행, j = 별, n = 공백

i	j	n
0	00	7
1	01	6
2	03	5
3	05	4
...	...	...
h-1	13	0

i=0	j=-1	n=0	기준값
i++	j++	n++	1씩 증가
i<h	j<2*i	n<(h-1)-i	식

풀이 : 최서연

# 문제1-2 다이아몬드 별짓기

```
Scanner sc = new Scanner(System.in);
while(true) { // 홀수만 입력 가능하게 판별위한 루프
    System.out.println("정수를 입력하세요");
    int n = sc.nextInt();
    if(n%2==0) { // 짝수의 경우
        System.out.println("홀수만 입력 가능합니다.");
    }else { // 홀수가 들어온 경우
        System.out.println("높이 : "+n);
        int i=0;
        while(i<n) { // 입력한 수 만큼 줄 출력
            if(i<(n+1)/2) { // 중심점 잡기
                int space=0; // 공백
                while(space<(n/2)-i) {
                    System.out.print(" ");
                    space++;
                }
                int star=0; // 별
                while(star<2*i+1) {
                    System.out.print("*");
                    star++;
                }
            }else {
                int space=0; // 공백
                while(space<=i-((n+1)/2)) {
                    System.out.print(" ");
                    space++;
                }
                int star=0; // 별
                while(star<(2*n-1)-(2*i)) {
                    System.out.print("*");
                    star++;
                }
            }
            System.out.println();
            i++;
        }
        break; // 완료 후 종료
    }
}
sc.close();
```

## 풀이 결과 출력

```
정수를 입력하세요
2
홀수만 입력 가능합니다.
정수를 입력하세요
7
      *
     ***
    *****
   *****
  *****
 *****
  *
```

```
정수를 입력하세요
9
높이 : 9
      *
     ***
    *****
   *****
  *****
 *****
 *****
  *****
 *****
  *
```

## 문제 풀이 과정

해당 별 짓기의 규칙 찾기

- 높이 7이라고 가정

1행 : 공백 3 별 1  
2행 : 공백 2 별 3  
3행 : 공백 1 별 5  
4행(분기점) : 별 7

- 증가 행
- 행이 증가함에 따라 공백은 1씩 줄어든고 별은 2씩 증가
- 공백 :  $7/2 - 1 = 3$  |  $7/2 - 2 = 2$  |  $7/2 - 3 = 1$  ->  $n/2 - i$
- 별 :  $2*0 + 1 = 1$  |  $2*1 + 1 = 3$  |  $2*2 + 1 = 5$  |  $2*3 + 1 = 7$  ->  $2*i + 1$

5행 : 공백 1 별 5  
6행 : 공백 2 별 3  
7행 : 공백 3 별 1

- 감소 행
- 공백 :  $5-4 = 1$  |  $6-4 = 2$  |  $7-4 = 3$  ->  $i - (n+1)/2$
- 별 :  $(2*7 - 1) - (2*4) = 5$  |  $(2*7 - 1) - (2*5) = 3$  |  $(2*7 - 1) - (2*6) = 1$   
->  $(2*n - 1) - (2*i)$

## 문제2 구매하기

```
3 class Buyer {
4     private int myMoney;
5     private int appleCnt;
6
7     // 모든 인자 생성자
8     public Buyer(int myMoney, int appleCnt) {
9         super();
10        this.myMoney = myMoney;
11        this.appleCnt = appleCnt;
12    }
13
14    public void payment(Seller seller, int money) { // 판매자에게 구매금액 전달 함수
15        this.myMoney -= money; // 전달한 금액만큼 보유 금액에서 차감
16        int applebuy = seller.receive(money); // 전달받은 사과개수
17        this.appleCnt += applebuy; // 현재 보유사과개수에 전달받은 사과개수 누적
18    }
19
20    // toString
21    @Override
22    public String toString() {
23        return "Buyer [money=" + myMoney + ", count=" + appleCnt + "];";
24    }
25
```

```
class Seller {
    private int myMoney;
    private int appleCnt;
    private int price;

    // 모든 인자 생성자
    public Seller(int myMoney, int appleCnt, int price) {
        super();
        this.myMoney = myMoney;
        this.appleCnt = appleCnt;
        this.price = price;
    }

    public int receive(int money) {
        this.myMoney += money; // 전달받은 금액만큼 보유 금액에 누적
        int applesold = money / price; // 전달할 사과개수 계산
        this.appleCnt -= applesold; // 내 사과개수에서 전달할 사과개수 차감
        return applesold; // 전달할 사과개수 리턴
    }

    // toString
    @Override
    public String toString() {
        return "Seller [money=" + myMoney + ", count=" + appleCnt + ", price=" + price + "];";
    }
}
```

```
public class Main {

    public static void main(String[] args) {

        Buyer 홍길동 = new Buyer(10000, 0);
        Buyer 철수 = new Buyer(20000, 0);

        Seller 사과장수 = new Seller(100000, 100, 1000);

        홍길동.payment(사과장수, 5000);
        철수.payment(사과장수, 12000);

        System.out.println(홍길동);
        System.out.println(철수);
        System.out.println(사과장수);
    }
}
```

### 풀이 결과 출력

```
Buyer [money=5000, count=5]
Buyer [money=8000, count=12]
Seller [money=117000, count=83, price=1000]
```

# 문제3 클로저 완성

```
closure.html X
closure.html > ...
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body>
    <script>
      function createCounter() {
        let count = 0;

        function increment() {
          // 코드 입력(호출될 때 마다 count에 1씩 증가)
          count++; // increment를 실행하면 count 변수에 +1씩 추가된다.
          return count;
        }
        return increment; // 코드 입력(increment 함수를 반환)
      }

      const counter = createCounter();
      console.log(counter()); // 출력 1
      console.log(counter()); // 출력 2
      console.log(counter()); // 출력 3
    </script>
  </body>
</html>
```

## 풀이 결과 출력

1	<u><a href="#">closure.html:22</a></u>
2	<u><a href="#">closure.html:23</a></u>
3	<u><a href="#">closure.html:24</a></u>

## 문제 풀이 과정

count ++; 코드를 추가해  
increment() 함수가 실행될 때 마다 count 변수에 +1을 해준다.

return increment; 코드를 추가해 함수를 반환한다.

## 설명

createCounter 함수 : count 변수를 0으로 초기화

함수 내부의 increment 함수를 정의하여 count++ 코드 실행

count 1씩 증가

increment 함수 반환

풀이 : 하태형