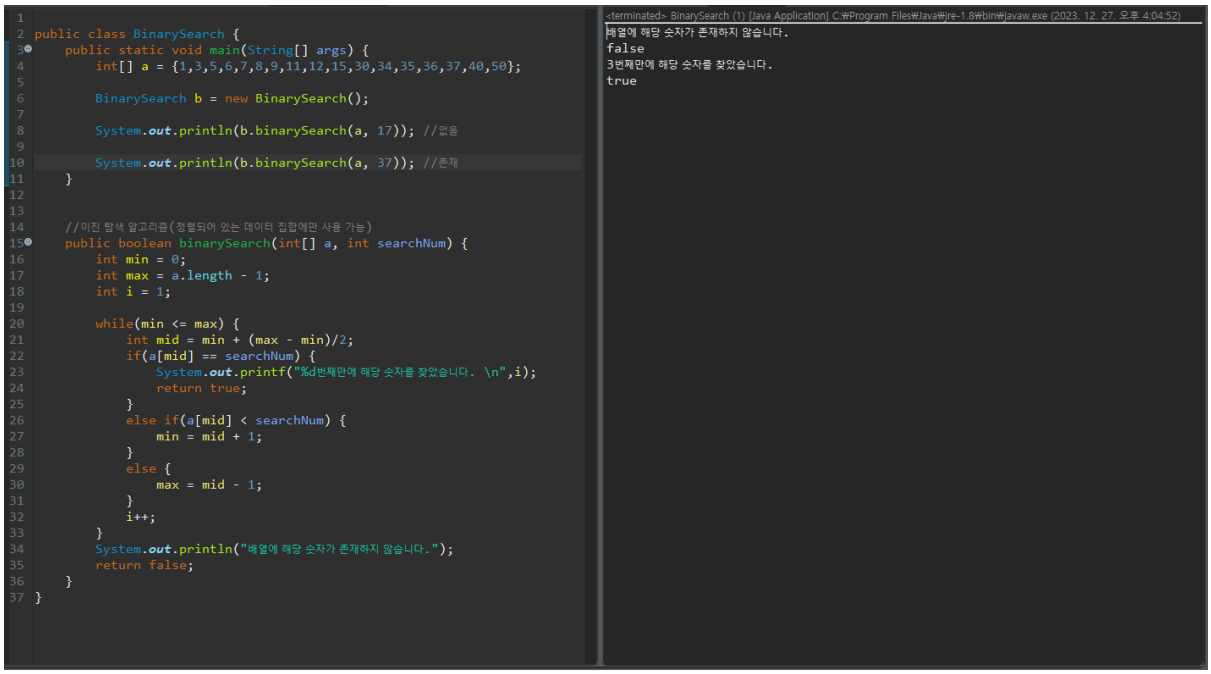
이진 탐색 알고리즘

➔ 정렬되어 있는 알고리즘에 사용 가능한 탐색 알고리즘

➔ 최소값과 최대 값의 중간 값을 기준으로 절차가 진행됨에 따라 중간 값이 찾고자 하는 값보다 작으면 최대 값을 중간 값으로 바꾸고 크면 최소 값을 중간 값으로 바꾸면서 중간 값을 새로 구하는 것을 반복하여 찾고자 하는 값을 찾는 알고리즘

시간 복잡도: $O(\log n)$



```java

public class BinarySearch {
    public static void main(String[] args) {
        int[] a = {1,3,5,6,7,8,9,11,12,15,30,34,35,36,37,40,50};

        BinarySearch b = new BinarySearch();

        System.out.println(b.binarySearch(a, 17)); //없음

        System.out.println(b.binarySearch(a, 37)); //존재
    }


    //이진 탐색 알고리즘(정렬되어 있는 데이터 집합에만 사용 가능)
    public boolean binarySearch(int[] a, int searchNum) {
        int min = 0;
        int max = a.length - 1;
        int i = 1;

        while(min <= max) {
            int mid = min + (max - min)/2;
            if(a[mid] == searchNum) {
                System.out.printf("%d번째만에 해당 숫자를 찾았습니다. \n",i);
                return true;
            }
            else if(a[mid] < searchNum) {
                min = mid + 1;
            }
            else {
                max = mid - 1;
            }
            i++;
        }
        System.out.println("배열에 해당 숫자가 존재하지 않습니다.");
        return false;
    }
}
```

```
<terminated> BinarySearch (1) [Java Application] C:₩Program Files₩Java₩jre-1.8₩bin₩javaw.exe (2023. 12. 27. 오후 4:04:52)
배열에 해당 숫자가 존재하지 않습니다.
false
3번째만에 해당 숫자를 찾았습니다.
true
```