자료구조 6주차

재귀함수

MMC 연구실

석사 과정 강민제

조교 소개

- 강민제
- 컴퓨터공학과 대학원 석사과정
- MMC연구실 (A1406)
- rkdalswp29@gmail.com

실습 수업 진행 방식

- 확인 문제 풀이

- 확인 문제를 해결한 학생은 검사 받고 퇴실

과제 설명

- 자료구조 수업은 Eclipse를 사용하여 코드를 작성합니다.
- 확인 문제 및 과제를 전부 해결하여 제출해주세요.
- 과제 제출 시 프로젝트 폴더를 압축해서 제출합니다.
- 과제의 채점은 프로젝트의 실행 결과를 기준으로 점수를 매깁니다.

확인문제 1 (factorial)

Package Name : recursion
Class Name : Factorial

```
package recursion;

public class Factorial {

    public static void main(String[] args) {
        System.out.println(factorial1(5));
        System.out.println(factorial2(5));
    }


    public static int factorial1(int number) {
        // 재귀함수를 이용하여 구현
    }

    public static int factorial2(int number) {
        // 반복문을 이용하여 구현
    }
}
```

```
<terminated> Factorial [Java Application]
120
120
```

확인문제 (factorial)

- $n = 0$: 1
- $n \geq 1$: $n \cdot (n-1) \cdot \ldots \cdot 2 \cdot 1 = n \cdot (n-1)!$

```
factorial(n)
        if  (n ≤ 1) then return 1
         else return (n · factorial(n –1));
 end factorial()
```

확인문제 2 (Binarysearch)

Package Name : recursion
Class Name : Binarysearch

```
public class BinarySearch {

    public static void main(String[] args) {
        int array1 [] = {1, 6, 13, 41, 45, 68, 70, 74, 81, 100};
        int array2 [] = {100, 68, 13, 41, 45, 6, 70, 74, 81, 1};
        System.out.println("array1에서 68의 위치 : " + search(array1, 68));
        System.out.println("array2에서 68의 위치 : " + search(array2, 68));
    }

    public static int search(int a [], int key) {
        // 배열이 정렬되어있는지 확인
        // 배열이 정렬되어있지 않다면 -1 리턴

        // binarySearch 메소드 호출
        // return binarySearch(???);

    }


    private static int binarySearch(int array [], int key, int left, int right) {
        // 재귀 알고리즘을 이용해 binary search 구현
    }


}
```

```
<terminated> BinarySearch [Java Application
array1에서 68의 위치 : 5
ERROR : 배열이 정렬되어 있지 않습니다.
array2에서 68의 위치 : -1
```

확인문제 2 (Binarysearch)

- key = a[mid] : 탐색 성공, return mid
- key < a[mid] : a[mid]의 왼편에 대해 이진탐색
- key > a[mid] : a[mid]의 오른편에 대해 이진탐색

확인문제 2 (Binarysearch)

```
binsearch(a[], key, left, right)
if (left ≤ right) then {
mid ← (left + right) / 2;
        case {
                key = a[mid] : return (mid);
                key < a[mid] : return (binsearch(a, key, left, mid - 1));
                key > a[mid] : return (binsearch(a, key, mid + 1, right));
        }
    }
    else return -1;
end binsearch()
```

확인문제 3 (fibonacci)

Package Name : recursion
Class Name : Fibo

```
public class Fibo {
    public static final int MAX_N = 10;
    public static void main(String[] args) {
        for(int i = 0; i <= MAX_N; i++)
            System.out.println(fib(i));

        System.out.println("------------------------------");

        for(int i = 0; i <= MAX_N; i++)
            System.out.println(fibIter(i));
    }

    public static long fib(int n) {
        // 재귀함수를 이용해 n번째 피보나치 수열의 값 리턴
    }

    public static long fibIter(int n) {
        // 반복문을 이용해 n번째 피보나치 수열의 값 리턴
    }


}
```

```
<terminated> Fibo [Java Application]
0
1
1
2
3
5
8
13
21
34
55
------------------------------
0
1
1
2
3
5
8
13
21
34
55
```

확인문제 3 (fibonacci)

피보나치 수 :

수학에서 피보나치 수는 첫째 및 둘째 항이 1이며 그 뒤의 모든 항은 바로 앞 두 항의 합인 수열이다. 처음 여섯 항은 각각 1, 1, 2, 3, 5, 8이다. 편의상 0번째 항을 0으로 두기도 한다.
위키백과

의견

과제 1 (Palindrome)

Package Name : recursion
Class Name : Palindrome

- **재귀함수를 이용**해 Palindrome인지 판단하는 메소드 구현

```
public class Palindrome {

    public static void main(String[] args) {
        System.out.println("abba : " + isPalin("abba"));
        System.out.println("abcba : " + isPalin("abcba"));
        System.out.println("abba : " + isPalin("accba"));
    }
    private static boolean isPalin(String s, int j, int k) {

    }

    public static boolean isPalin(String s) {
        return isPalin();
    }

}
```

```
<terminated> Palindrome [Java Application]
abba : true
abcba : true
abba : false
```

과제 1 (Palindrome)

- Palindrome : “eye”, “kayak”처럼 거꾸로 읽어도 제대로 읽는 것과 같은 문자열

- Palindrome인지 확인하는 방법

  - 반복문 혹은 재귀함수를 이용해 문자를 비교

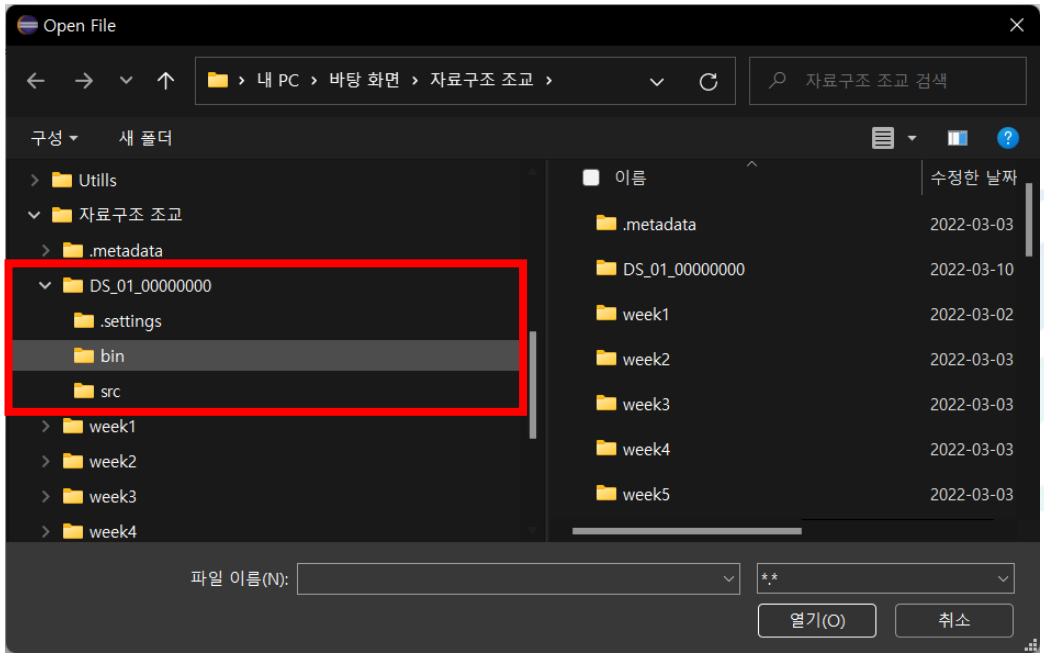
  “kayak” → “kayak” → “kayak” → O

  “apple” → X

- 문자열에서 문자를 하나 가져오는 법

```
String str = “abba”;
str.charAt(0); // “a”
str.charAt(1); // “b”
```

과제 제출 방법



- 프로젝트 폴더를 압축하여 제출

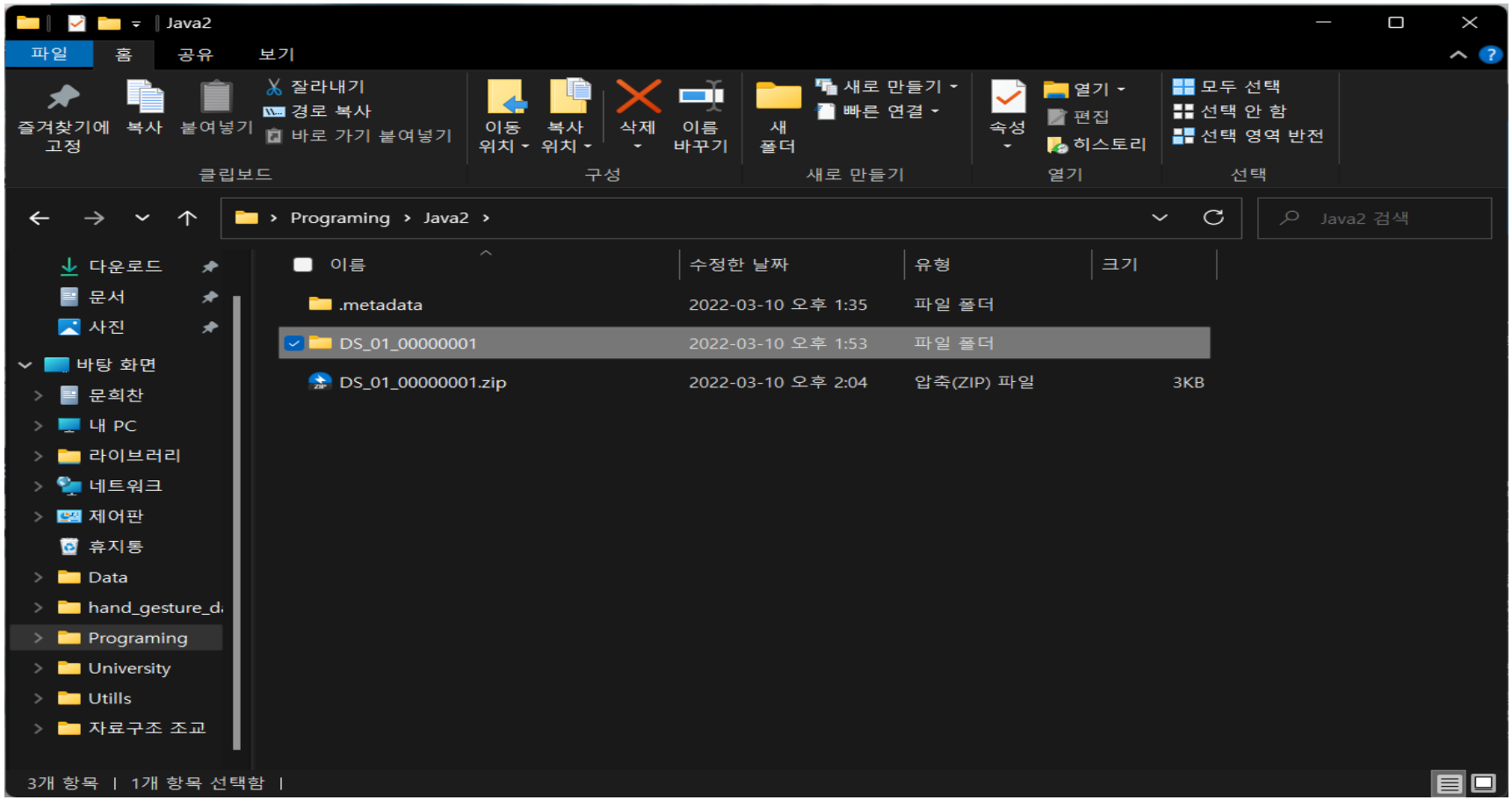
- 프로젝트이름 : DS_(주차)_(학번)

예) DS_06_00000000

- *.java파일만 제출하면 안됩니다.

제출양식을 반드시 지켜주세요

과제 제출 방법



- 반드시 **프로젝트 폴더를 압축**하여 제출