

모바일 프로그래밍

프로그래밍 과제 #2

제출일 : 2018. 10. 14.

학과 : 컴퓨터정보공학부

학번 : 2014722075

이름 : 이동준

과제결과요약

	결과	비고	자체평가
1번	검증완료		3점
2번	검증완료		3점
3번	검증완료		3점
4번	검증완료		3점
5번	검증완료		3점
6번	검증완료		3점

* 결과 - "검증완료, 디버깅필요, 미구현" 중 하나를 기재할 것.

* 비고 - 구체적인 사유를 요약해서 기재할 것. (위의 예시 참고)

* 자체평가 기준 - 3점: 검증 완료, 2점: 일부 버그 존재, 0점: 컴파일에러나 미구현

1번	결과: 검증 완료	자체평가 점수: 3점
----	-----------	-------------

- 코드

```

public class A2_1 {
    public static void main(String[] args) { //main method
        System.out.println("Student ID : 2014722075"); //print ID
        System.out.println("Name : 이동준"); //print Name

        //for receive
        Scanner keyboard = new Scanner(System.in);
        System.out.print("Enter number of data points: ");
        //Enter value
        int n = keyboard.nextInt(); //input number
        double[] arr = new double[n]; //make array
        double sum = 0;
        for(int i = 0 ; i<n ; i++)
        {
            System.out.print("[ "+i+" ]:");
            arr[i] = keyboard.nextDouble(); //enter values
            sum += arr[i]; //calculate sum of values
        }
        Arrays.sort(arr); //sort array
        double mean = sum/n; //calculate mean
        //print Result
        System.out.println("< RESULT >");
        String result = String.format("%.6f", mean);
        System.out.println("Statistical Information");
        System.out.println("=====");
        //print mean since ~.xxxxxx
        System.out.println("Arithmetic mean:"+result);
        //print median using Median method
        System.out.println("Median:"+Median(arr,n));
        //print mode using Mode method
        System.out.println("Mode:"+Mode(arr,n));
        //print Standard deviation using Standard method
        System.out.println("Standard deviation:"+Standard(arr,n,mean));
        keyboard.close();
    }

    public static double Median(double[] arr,int n)
    { //calculate median method
        double median = 0;;
        if(n%2 == 0) //if size of array is even number
            median = (float)(arr[n/2]+arr[n/2-1])/2;
        else //else
            median = arr[n/2];

        return median; //return result
    }
}

```

```

public static int Mode(double[] arr,int n)
{
    //find mode method
    int mode = 0;int mcount = 0;int count = 0;
    for(int i = 0;i<n;i++)
    {
        count = 0;
        for(int j = 0; j<n ; j++)
        {
            //counting
            if(arr[i] == arr[j])
                count++;
        }
        if(count > mcount)
        {
            //if need to change mode
            mcount = count;
            mode = (int)arr[i];
        }
    }
    return mode;//return result
}

public static String Standard(double[] arr, int n, double mean)
{
    //calculate Standard deviation method
    double S1 = 0;
    double S2 = 0;
    //difference between the mean and sum of all
    for(int i = 0 ;i < n ; i++)
        S1 += Math.pow((float)arr[i]-mean, 2);
    S2 = Math.sqrt(S1/(n-1)); //calculate square value
    String result = String.format("%.6f", S2);
    return result;//return result
}
}

```

- 실행 결과

```

Student ID : 2014722075
Name : 이동준
Enter number of data points: 12
[0]:23
[1]:23
[2]:33
[3]:23
[4]:43
[5]:43
[6]:43
[7]:41
[8]:23
[9]:23
[10]:32
[11]:23
< RESULT >
Statistical Information
=====
Arithmetic mean:31.083333
Median:27.5
Mode:23
Standard deviation:9.149847

Student ID : 2014722075
Name : 이동준
Enter number of data points: 6
[0]:1
[1]:2
[2]:3
[3]:4
[4]:5
[5]:3
< RESULT >
Statistical Information
=====
Arithmetic mean:3.000000
Median:3.0
Mode:3
Standard deviation:1.414214

```

* 입력을 통해 number of data points을 할당한 뒤 입력한 수 만큼 integer 수를 입력 받는다. 입력을 마친 후 mean, median, mode, standard deviation를 계산한다. mean의 경우에는 전체의 수를 더한 값을 전체 수의 개수로 나누어 준다. 나머지 세 항목에 대해서는 method를 따로 생성하여 계산을 진행한다. median method는 입력 받는 수가 짝수일 경우에는 중앙 두 index의 value를 더해 절반 값을 취하였고, 홀수일 경우에는 중앙 index의 value를 취한다. mode method는 for문 두 개를 이용해 배열 전체를 순회하면서 count를 세준다. 가장 큰 count가 발생할 경우 그 index가 가지고 있는 value를 취한다. 마지막으로 Standard method는 표준편차를 구하는 공식에 의해 Math function을 이용하여 계산하였다.

2번	결과: 검증 완료	자체평가 점수: 3점
----	-----------	-------------

- 코드

```
public class A2_2 {
    public static void main(String[] args) { //main method
        System.out.println("Student ID : 2014722075"); //print ID
        System.out.println("Name : 이동준"); //print Name

        //print Result
        System.out.println("< RESULT >");
        System.out.println("add(1,2) -> "+add(1,2));
        System.out.println("add(1,2,3) -> "+add(1,2,3));
        System.out.println("add(1.4, 4.5) -> "+add(1.4,4.5));
    }
    //Method using overloading
    public static int add(int a,int b)
    { return a+b; } //2 integer addition
    public static int add(int a,int b, int c)
    { return a+b+c; } //3 integer addition
    public static double add(double a,double b)
    { return a+b; } //2 double addition
    public static double add(double a,double b, double c)
    { return a+b+c; } //3 double addition
}
```

- 실행 결과

```
Student ID : 2014722075
Name : 이동준
< RESULT >
add(1 , 2) -> 3
add(1 , 2 , 3) -> 6
add(1.4 , 4.5) -> 5.9
add(4 , 5 , 6) -> 15
add(1.1 , 2.2 , 3.3) -> 6.6
add(100 , 50) -> 150
```

* add method를 생성하여 2~3개의 수를 덧셈하는 프로그램을 작성한다. 만약 add의 인자가 int형 변수들이라면 결과도 int, double형이면 결과도 double으로 출력된다. add method는 overloading의 형태를 띄고 있으며, overloading이라함은 method의 이름은 동일하나 parameter가 다른 함수를 의미한다. 즉 parameter가 int형 일 때와 double형 일 때의 실행되는 함수는 다르지만 call하는 method의 이름은 같다.

3번	결과: 검증 완료	자체평가 점수: 3점
----	-----------	-------------

- 코드

```
public class A2_3 {
    public static void main(String[] args) { //main method
        System.out.println("Student ID : 2014722075"); //print ID
        System.out.println("Name : 이동준"); //print Name

        //for receive
        Scanner keyboard = new Scanner(System.in);
        double[] arr = new double[100]; //make array
        System.out.print("Enter the numbers : ");
        int i = 0; int c = 0;
        //input numbers if input is 0, don't need to input
        for(i = 0; i<100 ; i++)
        {
            arr[i] = keyboard.nextDouble();
            if(arr[i] == 0)
                break;
        }

        double[] arr2 = new double[i]; //make new array
        //copy numbers, new array has only number input by user
        for(c = 0 ; c < i ; c++)
            arr2[c] = arr[c];
        Arrays.sort(arr2); //sort array
        //print Result
        System.out.println("< RESULT >");
        for(int j = 1 ; j < c ; j++)
        {
            if(arr2[j-1] == arr2[j] && arr2[j] != arr2[j+1])
                System.out.print(arr2[j]+" ");
        }

        keyboard.close();
    }
}
```

- 실행 결과

```
Student ID : 2014722075
Name : 이동준
Enter the numbers : 10 20 30 40 50 10 20 30 60 70 0
< RESULT >
10.0 20.0 30.0
```

```
Student ID : 2014722075
Name : 이동준
Enter the numbers : 1.1 2.2 3.3 1 2 3 1.1 2.2 4.4 4 0
< RESULT >
1.1 2.2
```

* 입력을 통해 최대 100개까지의 양의 실수를 입력 받으며 만약 입력된 값이 0일 경우에는 입력을 중단한다. 크기가 100인 double형태의 array를 선언하여 입력한 수들을 임시 저장하고 0을 제외한 실제 입력된 수들만 존재하는 입력된 수의 개수만큼 크기를 가지는 배열을 만든다. Arrays.sort를 이용해 오름차순 sort를 진행하고 중복으로 입력된 수에 대해서만 출력을 한다.

4번	결과: 검증 완료	자체평가 점수: 3점
----	-----------	-------------

- 코드

```

abstract class Electric
{
    //abstract class Electric
    //instance
    void nameofClass()
    {
        //print name of class method
        System.out.println("\n==Electric Class==");
    }
    abstract void printInstance(); //abstract method
    void printMethods()
    {
        //print methods
        System.out.println("Class has nameofClass(), printInstance(), printMethods()");
    }
}

class Smartphone extends Electric
{
    //class Smartphone extends Electric Class
    String S = "Galaxy S7"; //member value
    void printInstance()
    {
        //print member instance method
        System.out.println("My Smartphone is "+S);
    }
}

class laptop extends Electric
{
    //class laptop extends Electric Class
    String S = "LG Gram"; //member value
    void printInstance()
    {
        //print member instance method
        System.out.println("My laptop is "+S);
    }
}

abstract class Book
{
    //abstract class Book
    //instance
    void nameofClass()
    {
        //print name of class method
        System.out.println("\n==Book Class==");
    }
    abstract void printInstance(); //abstract method
    void printMethods()
    {
        //print methods
        System.out.println("Class has nameofClass(), printInstance(), printMethods()");
    }
}

class Book1 extends Book
{
    //class Book1 extends Book
    String S = "Bible of Math"; //member value
    void printInstance()
    {
        //print member instance method
        System.out.println("My Book1 is "+S);
    }
}

class Book2 extends Book
{
    String S = "Master of JAVA"; //member value
    void printInstance()
    {
        //print member instance method
        System.out.println("My Book2 is "+S);
    }
}

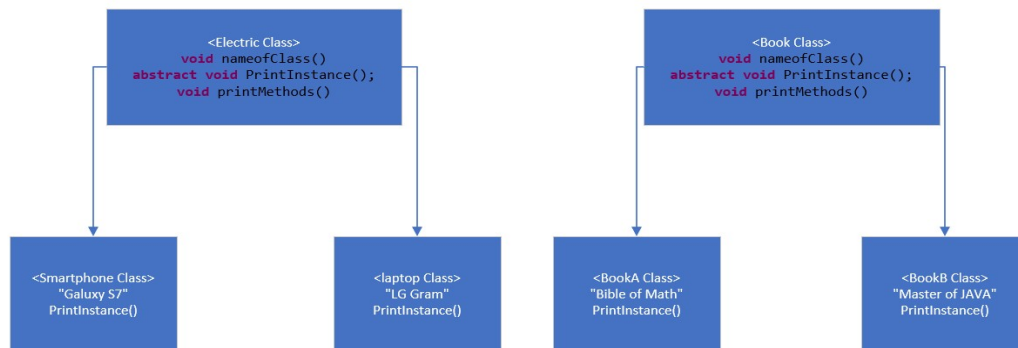
```

```

public class A2_4 {
    public static void main(String[] args) { //main method
        System.out.println("Student ID : 2014722075"); //print ID
        System.out.println("Name : 이동준"); //print Name

        //make object and make new type
        Electric S = new Smartphone();
        Electric L = new laptop();
        Book A = new Book1();
        Book B = new Book2();
        //print Result
        System.out.println("< RESULT >");
        //call each methods
        S.nameOfClass();
        S.printInstance();
        S.printMethods();
        L.nameOfClass();
        L.printInstance();
        L.printMethods();
        A.nameOfClass();
        A.printInstance();
        A.printMethods();
        B.nameOfClass();
        B.printInstance();
        B.printMethods();
    }
}

```



- 실행 결과

Student ID : 2014722075

Name : 이동준

< RESULT >

==Electric Class==

My Smartphone is Galuxy S7

Class has nameofClass(), printInstance(), printMethods()

==Electric Class==

My laptop is LG Gram

Class has nameofClass(), printInstance(), printMethods()

==Book Class==

My Book1 is Bible of Math

Class has nameofClass(), printInstance(), printMethods()

==Book Class==

My Book2 is Master of JAVA

Class has nameofClass(), printInstance(), printMethods()

* abstract Class Electric과 Book을 만든 뒤 각 클래스에 포함되어 있는 항목을 class로 만들어 상속을 시킨다. Class의 이름을 출력하는 method와 Methods를 출력하는 method는 abstract class에 정의하여 상속 받은 모든 class의 object에서 동일하게 출력하도록 하였고, 각 instance를 출력하는 method는 상속을 받은 class에서 정의하여 각 object에 맞게 출력을 하도록 하였다.

5번	결과: 검증 완료	자체평가 점수: 3점
----	-----------	-------------

- 코드

```

public class A2_5 {
    public static void main(String[] args) { //main method
        System.out.println("Student ID : 2014722075"); //print ID
        System.out.println("Name : 이동준"); //print Name

        //make ArrayList using Integer
        ArrayList<Integer> a = new ArrayList<Integer>();
        for(int i = 0; i<10; i++)
        { //insert 10 numbers
            while(true)
            { //it is function to avoid duplication
                int temp = (int)(Math.random()*20)+1;
                if(!a.contains(temp))
                {
                    a.add(temp);
                    break;
                }
            }
        }
        a.sort(null); //array sort
        System.out.println("List: "+a);
        System.out.print("Input your command: ");
        String c = ""; //input command
        Scanner keyboard = new Scanner(System.in);

        while(true)
        {
            c = keyboard.next(); //input command
            if(c.equals("a"))
            { //if command is a
                System.out.print("Input number to add: ");
                //input add number
                int addnum = keyboard.nextInt();
                ADD(a,addnum); //call ADD method
            }
            else if(c.equals("r"))
            { //if command is r
                System.out.print("Input number to remove: ");
                //input remove number
                int remnum = keyboard.nextInt();
                REMOVE(a,remnum); //call REMOVE method
            }
            else if(c.equals("s"))
            { //if command is s
                System.out.print("Input number to search: ");
                //input search number
                int seanum = keyboard.nextInt();
                SEARCH(a,seanum); //call SEARCH method
            }
            else //if command is e
                break; //end
            a.sort(null); //array sort
            System.out.println("List: "+a);
            System.out.print("Input your command: ");
            c = null;
        }
        System.out.println("<END> ");
        keyboard.close();
    }
}

```

```

    public static void ADD(ArrayList<Integer> a,int addnum)
    {
        //insert new number in ArrayList
        a.add(addnum);
    }
    public static void REMOVE(ArrayList<Integer> a,int remnum)
    {
        //remove number in ArrayList
        if(a.indexOf(remnum) == -1) //if not exist input number
            System.out.println("Not exist Number in array");
        else //if exist input number
            a.remove(a.indexOf(remnum));
    }
    public static void SEARCH(ArrayList<Integer> a,int seanum)
    {
        //search number in ArrayList
        if(a.indexOf(seanum) == -1) //if not exist input number
            System.out.println("Not exist Number in array");
        else //if exist input number
            System.out.println("Index of "+seanum+" is "+a.indexOf(seanum));
    }
}

```

- 실행 결과

```

Student ID : 2014722075
Name : 이동준
List: [3, 6, 7, 9, 10, 12, 15, 16, 17, 19]
Input your command: a
Input number to add: 1
List: [1, 3, 6, 7, 9, 10, 12, 15, 16, 17, 19]
Input your command: a
Input number to add: 18
List: [1, 3, 6, 7, 9, 10, 12, 15, 16, 17, 18, 19]
Input your command: r
Input number to remove: 19
List: [1, 3, 6, 7, 9, 10, 12, 15, 16, 17, 18]
Input your command: s
Input number to search: 1
Index of 1 is 0
List: [1, 3, 6, 7, 9, 10, 12, 15, 16, 17, 18]
Input your command: e
<END>

```

* ArrayList를 통해 10개의 정수(1~20)를 랜덤으로 insert한다. 오름차순으로 sort를 진행하고 command를 입력 받는다. a는 ADD method를, r은 REMOVE method를, s은 SEARCH method를 실행하도록 구현하였고, 종료를 위해 다른 커맨드를 입력 시 END의 기능을 추가하였다. ADD method는 ArrayList의 add 명령어를 통해 입력 받은 수를 array에 추가한다. REMOVE method는 ArrayList의 remove 명령어를 통해 입력 받은 수를 array에서 제거한다. 만약 array에 존재하지 않은 수일 경우 에러를 출력한다. SEARCH method는 ArrayList의 indexOf 명령어를 통해 찾으려는 수가 존재하는 배열의 index를 출력한다. 만약 array에 존재하지 않은 수일 경우 에러를 출력한다.


```

boolean FindPath(char[] word)
{
    //find path method
    int n = word.length; int[] f = new int[n];
    int i = 0; int count = 0;
    f[0] = Find(word[0]);
    if(f[0] == -1)
        return false;
    i++; int temp = -1; int bfs = -1;
    visited[f[0]] = true;
    while(true)
    {
        if(i == n)
            break;
        f[i] = Find(word[i]);
        if(i == 0)
        {
            if(f[0] == -1 || bfs == f[0])
                return false;
            resetVisit();
            visited[f[0]] = true;
            i++;
            continue;
        }
        if(f[i] == -1)
        {
            if(temp != -1)
            {
                visited[temp] = false;
                temp = -1;
            }
            i--; temp = f[i];
            if(i == 0 && count == 0)
            {
                bfs = f[i]; count++;
                resetVisit();
            }
            visited[f[i]] = true;
        }

        else
        {
            if(CanNext(f[i-1],f[i]))
            {
                visited[f[i]] = true;
                if(temp != -1 && temp != f[i])
                {
                    visited[temp] = false;
                    temp = -1;
                }
                i++;
            }
            else
            {
                visited[f[i]] = true;
                temp = f[i];
            }
        }
    }
    return true; //exist
}

```



```

        boolean CanNext(int i,int j)
        {
            //check link of grid method
            if(i == 0) {
                if(j ==1||j==3||j==4)
                    return true;
            }
            else if(i == 1){
                if(j==0||j==2||j==3||j==4||j==5)
                    return true;
            }
            else if(i == 2){
                if(j==1||j==4||j==5)
                    return true;
            }
            else if(i == 3){
                if(j==0||j==1||j==4||j==6||j==7)
                    return true;
            }
            else if(i == 4){
                return true;
            }
            else if(i == 5){
                if(j==1||j==2||j==4||j==7||j==8)
                    return true;
            }
            else if(i == 6){
                if(j==3||j==4||j==7)
                    return true;
            }
            else if(i == 7){
                if(j==3||j==4||j==5||j==6||j==8)
                    return true;
            }
            else if(i == 8){
                if(j==4||j==5||j==7)
                    return true;
            }
            return false;
        }

        boolean Exist(char[] word)
        {
            //check if path is exist or not
            if(FindPath(word))
                return true;
            return false;
        }

        void Manager()
        {
            //game manager
            Scanner keyboard = new Scanner(System.in);
            String Word;
            while(true)
            {
                System.out.print("Input word: ");
                Word = keyboard.next(); //input word
                if(Word.length() < 3)
                {
                    //if word's size is < 3
                    System.out.println("!!Input word is at least three letters!!");
                    continue; //error
                }
                if(Word.equals("END")) //if input is END
                    break; //end
                if(Word.equals("GRID")) //if input is GRID
                {
                    ShowGrid(); //call show grid method
                    continue;
                }
                //start game
                int size = Word.length();
                char[] w = new char[size];
                w = Word.toCharArray();
                if(Exist(w) == true)
                    System.out.println("Exist");
                else
                    System.out.println("Non-Exist");
                resetVisit();
            }
            System.out.println("END OF GAME");
            keyboard.close();
        }
    }
}

```


- 실행 결과

<pre> Student ID : 2014722075 Name : 이동준 ----- P X K ----- N V W ----- R Q C ----- Input word: P XK Exist Input word: N VRQ Exist Input word: CWP Non-Exist Input word: GRID ----- P X K ----- N V W ----- R Q C ----- Input word: END END OF GAME </pre>	<pre> Student ID : 2014722075 Name : 이동준 ----- K V G ----- I M S ----- S O C ----- Input word: SIK Exist Input word: SOS Exist Input word: CSK Non-Exist Input word: OSK Non-Exist Input word: SOCS Exist Input word: KVGs Exist Input word: END END OF GAME </pre>
---	---

* 크기가 9인 배열 grid와 visited을 가지고 있는 Boggle Class를 생성하여 각 기능에 맞도록 method를 설정한다. 생성자, 게임매니저, findpath 등 게임에 필요한 method를 설정한다. main에 선 Boggle object를 생성하고 Manager method를 통해 게임을 진행한다. 총 9개의 index를 linked list처럼 사용을 하고 있으며 연결상태에 따라 연결된 길을 판단하는 method를 생성하여 진행 방향을 판단한다. 크기 9의 GRID를 난수로 생성하기 때문에 중복값에 대한 처리가 필요하다. 각 중복값에 대한 path를 판단하기 위해 다양한 방법을 시도하였다. 과도한 중복값에 의한 에러가 존재할 수 있지만 현실성을 고려하였을 때 현실적인 Case에서는 모두 동작을 완료한다.

Input word를 통해 찾고자 하는 word를 입력하고 Exist 또는 Non-Exist를 출력한다. GRID를 입력할 시 구성되어 있는 GRID를 출력하며, 게임 종료를 위해 END를 입력 시 종료하게 설계하였다.