

# 소프트웨어 프로젝트 Report 3

과 목 명 : 소프트웨어 프로젝트

과 제 명 : Circle 외 2

사용언어 : 자바

학 과 : 심리학과

학 번 : 20141963

이 름 : 박승완

제 작 일 : 20190509

담당교수 : 이남규교수님

1. 중심을 나타내는 정수  $x$ ,  $y$ 와 반지름  $radius$  필드를 가지는 `Circle` 클래스를 작성하라. `equals()` 메소드는 두개의 `Circle` 객체의 중심이 같으면 같은 것으로 판별하도록 한다.

힌트) 객체를 출력하기 위한 `toString()`과 비교를 위한 `equals()` 메소드를 만들어야 함.

```
public class CircleApp {  
  
    public static void main(String[] args) {  
        Circle a = new Circle(2,3,5); // 중심 (2,3)에 반지름 5인 원  
        Circle b = new Circle(2,3,30); // 중심 (2,3)에 반지름 30인 원  
        System.out.println("원 a : " + a);  
        System.out.println("원 b : " + b);  
        If (a.equals(b))  
            System.out.println("같은 원");  
        else  
            System.out.println("서로 다른 원");  
    }  
}
```

실행결과:

```
원 a : Circle(2,3) 반지름 5  
원 b : Circle(2,3) 반지름 30  
같은 원
```

**생성할 것들 :**

`Circle` class ( $x$ ,  $y$ ,  $radius$ 를 변수 필드에 가짐, 생성자함수에 인자 3개를 받아서 각각을 초기화),

`toString` 메서드 ( "원 a : Circle(2,3) 반지름 5" 이렇게 출력하게 함)

`equals` 메서드 (두 원의 중심이 같은지를 체크하여 `boolean`을 리턴)

**해결 방법 :**

`Circle` 클래스와 그 안의 메서드들을 `public`으로 지정해서 `CircleApp`이 같은 패키지에 없더라도 사용할 수 있게 해준다.

위의 "생성할 것들"에 있는 것들을 생성해준다.

교수님께서 올려주신 `CircleApp`클래스를 실행한다.

실행결과 :

Circle class

```
New Java Class  ad.jav  GradeCal.java  Rainfall.java  MyString.java  Circle.java  Circle
1 package report3;
2 /* Circle class
3  * 생성자에 인자를 3 개 받아 각각 원 중심의 x, y 그리고 반지름을 저장한다.
4  *
5  * 객체를 이용해 출력을 할 시 출력할 내용을 toString() 메서드를 이용해 지정해준다.
6  *
7  * 두 원의 중심이 같은지 여부를 boolean으로 확인해주는 equals() 메서드도 만든다.
8  */
9 import java.lang.String;
10
11 public class Circle {
12
13     int x, y, radius;
14     //생성
15     public Circle(int a, int b, int c) {
16         this.x = a;
17         this.y = b;
18         this.radius = c;
19     }
20
21     //객체를 이용해 출력시 텍스트 지정.
22     public String toString() {
23         return "Circle(" + this.x + "," + this.y + ") 반지름" + this.radius;
24     }
25
26     //두 원의 중심이 같은지 여부.
27     public boolean equals(Circle circle) {
28         boolean result;
29         if(this.x == circle.x && this.y == circle.y)
30             result = true;
31         else
32             result = false;
33         return result;
34     }
35 }
36
37
```

CircleApp class 및 실행결과

```
RandAndRead.jav  GradeCal.java  Rainfall.java  MyString.java  Circle.java  CircleApp.java  MyString
1 package report3;
2 /* 두 원의 정보를 나타내고, 같은지 다른지를 표시하는 프로그램.
3  * 두 원의 객체를 만들고 각 원에 중심과 반지름을 인자로 준다.
4  * 원의 정보를 출력한다.
5  * 원이 같은지 비교를 한다.
6  */
7 public class CircleApp {
8
9     public static void main(String[] args) {
10         Circle a = new Circle(2,3,5); // 중심 (2,3)에 반지름 5인 원
11         Circle b = new Circle(2,3,30); // 중심 (2,3)에 반지름 30인 원
12         System.out.println("원 a : " + a);
13         System.out.println("원 b : " + b);
14         if(a.equals(b))
15             System.out.println("같은 원");
16         else
17             System.out.println("서로 다른 원");
18     }
19 }
20
```

Problems Javadoc Declaration Console <terminated> CircleApp (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0\_202.jdk/Contents/Home/bin/java (2019. 5. 7. 오후 7:49:55)

```
원 a : Circle(2,3) 반지름5
원 b : Circle(2,3) 반지름30
같은 원
```

2. **MyString** 클래스의 구현: **String** 클래스와 유사한 나만의 문자열 클래스로써 **MyString** 클래스를 만들고, **MyStringEx** 클래스를 이용하여 실행되도록 구현하시오.

확인용 **MyStringEx** class : (교수님께서 첨부해주심)

```
public class MyStringEx {  
  
    public static void main(String[] args) {  
        MyString s = new MyString(new char[] { 'a', 'B', 'c', 'D' });  
        MyString s1 = new MyString(new char[] { 'a', 'B', 'c', 'D' });  
        MyString s2 = new MyString(new char[] { 'a', 'b', 'c', 'd' });  
  
        System.out.print("s = ");  
        MyString.println(s); // aBcD  
        System.out.print("s1 = ");  
        MyString.println(s1); // aBcD  
        System.out.print("s2 = ");  
        MyString.println(s2); // abcd  
  
        System.out.println("s.length = " + s.length()); //4  
        System.out.println("s.equals(s1) = " + s.equals(s1)); // true  
        System.out.println("s.equals(s2) = " + s.equals(s2)); // false  
        System.out.println("s.charAt(1) = " + s.charAt(1)); // B  
  
        MyString s3 = s.toUpperCase();  
        System.out.print("s.toUpperCase() = ");  
        MyString.println(s3); // ABCD  
  
        MyString s4 = s.substring(1, 2);  
        System.out.print("s.substring(1, 2) = ");  
        MyString.println(s4); // Bc  
  
        System.out.print("MyString.valueOf(345) = ");  
        MyString.println(MyString.valueOf(345)); // 345  
  
        System.out.print("MyString.valueOf(true) = ");  
        MyString.println(MyString.valueOf(true)); // true  
    }  
}
```

**MyString** class에 만들 메서드들.

- \* 출력 (println) - **MyString**, int, **boolean**을 받을 수 있도록 오버로딩.
- \* 길이 (length)
- \* 같은지 여부 (equals)
- \* 해당 인덱스에 어떤 값이 있는지 (charAt)
- \* 대문자로 (toUpperCase)
- \* 인덱스로 substring만들기 (substring)
- \* 객체 생성 안하고 바로 들어가는 인자 반환하기 (valueOf)

## 해결방법 :

위에 명시한 메서드들을 MyString 클래스에 만들어줍니다.

toUppercase 메서드의 경우에는 `Character.toUpperCase(myString[i])`를 이용하고 for문을 이용하여 i를 인자로하여 myString의 크기만큼 반복하는 방법으로 대문자화합니다.

다른 메서드들은 쉽게 구현이 가능합니다.

## 실행 결과 :

MyString class

```
1 package report3;
2 /* MyString클래스를 구현하기 : String 클래스와 유사한 나만의 문자열 클래스.
3 *
4 * 출력 (println) - MyString, int, boolean을 받을 수 있도록 오버로딩.
5 * 길이 (length)
6 * 같은지 여부 (equals)
7 * 해당 인덱스에 어떤 값이 있는지 (charAt)
8 * 대문자로 (toUppercase)
9 * 인덱스로 substring만들기 (substring)
10 * 객체 생성 안하고 바로 들어가는 인자 반환하기 (valueOf)
11 */
12 public class MyString {
13     //변수 설정.
14     public char[] mystring;
15     //생성자.
16     public MyString(char[] input) {
17         this.mystring = input;
18     }
19     //println 메서드 (각각 MyString, int, boolean을 인자로 받는다.)
20     public static void println(MyString input) {
21         System.out.println(input.mystring);
22     }
23     public static void println(int input) {
24         System.out.println(input);
25     }
26     public static void println(boolean input) {
27         System.out.println(input);
28     }
29     //length 메서드 (길이 반환)
30     public int length() {
31         return mystring.length;
32     }
33     //equals 메서드 (같은지 여부 반환)
34     public boolean equals(MyString check) {
35         String str_input = String.valueOf(mystring);
36         String str_check = String.valueOf(check.mystring);
37         if(str_input.equals(str_check))
38             return true;
39         else
40             return false;
41     }
42     //charAt 메서드 (인자로 받은 인덱스값에 해당하는 값을 반환)
43     public char charAt(int index) {
44         return mystring[index];
45     }
46     //toUppercase 메서드 (대문자로 변환 후 반환)
47     public MyString toUppercase() {
48         char [] temp_char = new char[mystring.length];
49         for(int i = 0; i < mystring.length; i++) {
50             temp_char[i] = Character.toUpperCase(mystring[i]);
51         }
52         MyString temp = new MyString(temp_char);
53         return temp;
54     }
55     //substring 메서드 (받은 두 인덱스 사이의 값을 출력)
56     public MyString substring(int ind1, int ind2) {
57         char [] temp_char = new char[(ind2 - ind1) + 1];
58         for(int i = 0; i < (ind2 - ind1) + 1; i++) {
59             temp_char[i] = mystring[ind1 + i];
60         }
61         MyString temp = new MyString(temp_char);
62         return temp;
63     }
64     //valueOf 메서드 (input으로 받은 값을 넘겨준다. - int, boolean)
65     public static int valueOf(int input) {
66         return input;
67     }
68     public static boolean valueOf(boolean input) {
69         return input;
70     }
71 }
```

MyStringEx class 실행 결과

```
Problems @ Javadoc Declaration C
<terminated> MyStringEx [Java Application] /Librar
s = aBcD
s1 = aBcD
s2 = abcd
s.length = 4
s.equals(s1) = true
s.equals(s2) = false
s.charAt(1) = B
s.toUppercase() = ABCD
s.substring(1, 2) = Bc
MyString.valueOf(345) = 345
MyString.valueOf(true) = true
```

## MyStringEx class

```
1 package report3;
2 /* MyString을 구동할 MyStringEx 클래스.
3 * MyString 객체 s, s1, s2에 각각 char배열로 초기화한 것을 생성.
4 * 출력 (println)
5 * 길이 (length)
6 * 같은지 여부 (equals)
7 * 해당 인덱스에 어떤 값이 있는지 (charAt)
8 * 대문자로 (toUpperCase)
9 * 인덱스로 substring만들기 (substring)
10 * 객체 생성 안하고 바로 들어가는 인자 반환하기 (valueOf)
11 */
12 public class MyStringEx {
13
14     public static void main(String[] args) {
15         MyString s = new MyString(new char[] {'a', 'B', 'c', 'D'});
16         MyString s1 = new MyString(new char[] {'a', 'B', 'c', 'D'});
17         MyString s2 = new MyString(new char[] {'a', 'b', 'c', 'd'});
18
19         System.out.print("s = ");
20         MyString.println(s); // aBcD
21         System.out.print("s1 = ");
22         MyString.println(s1); // aBcD
23         System.out.print("s2 = ");
24         MyString.println(s2); // abcd
25
26         System.out.println("s.length = " + s.length());
27         System.out.println("s.equals(s1) = " + s.equals(s1)); // true
28         System.out.println("s.equals(s2) = " + s.equals(s2)); // false
29         System.out.println("s.charAt(1) = " + s.charAt(1)); // B
30
31         MyString s3 = s.toUpperCase();
32         System.out.print("s.toUpperCase() = ");
33         MyString.println(s3); // ABCD
34
35         MyString s4 = s.substring(1, 2);
36         System.out.print("s.substring(1, 2) = ");
37         MyString.println(s4); // Bc
38
39         System.out.print("MyString.valueOf(345) = ");
40         MyString.println(MyString.valueOf(345)); // 325
41
42         System.out.print("MyString.valueOf(true) = ");
43         MyString.println(MyString.valueOf(true)); // true
44     }
45 }
46
```

3. 도시 이름, 위도, 경도 정보를 가진 Location 클래스를 작성하고, 도시이름을 키로 하는 HashMap<String, Location> 컬렉션을 만드시오. 사용자로부터 입력 받아 4개의 도시를 아래 실행 결과를 참고하여 실행하라. 그리고 도시의 이름으로 검색하는 프로그램을 작성하라.

실행결과:

도시, 경도, 위도를 입력하세요.

>> 서울, 37, 126

>> LA, 34, -118

>> 파리, 2, 48

>> 시드니, 151, -33

해시맵에 있는 모든 도시를 출력합니다.

서울, 37, 126

LA, 34, -118

파리, 2, 48

시드니, 151, -33

도시를 검색합니다.

도시이름 >> 부산

부산 없습니다.

도시이름 >> 파리

파리 2 48

도시이름 >> 종료

검색을 종료합니다.

## 해결방법 :

사용자로부터 도시 이름, 위도, 경도를 4번 입력 받아 도시이름을 키로하여 HashMap에 저장한다.

이때 위도, 경도는 하나의 문자열로 만들어주어 키에 해당하는 값으로 저장한다.

저장이 완료되면 HashMap에 있는 모든 정보(4개)를 출력한다.

그리고 검색기능을 구현한다.

검색기능 : 문자열을 입력 받고 그 문자열에 해당되는 키가 있을 시 키와 값을 출력.

없다면 입력 받은 값이 없다고 출력.

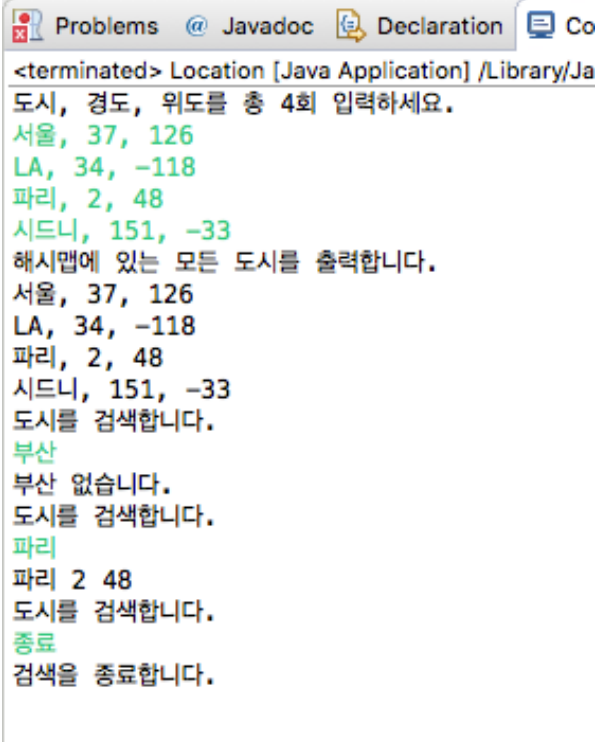
종료를 입력하면 종료되게 한다.

## 실행결과:

Location class

```
1 ② /* 도시 이름, 위도, 경도 정보를 가진 Location 클래스를 작성하고, 도시이름을 '키'로 하는 HashMap<String, Location> 컬렉션을 만드시오.
2  * 사용자로부터 입력 받아 4개의 도시를 아래 실행 결과를 참고하여 실행하라.
3  * 그리고 도시의 이름으로 검색하는 프로그램을 작성하라.
4  * 사용자로부터 이름 받아서 키로 쓰고, 위도 & 경도를 받아서 한 스트링으로 만들어서 값으로 넣자.
5  */
6 package report3;
7 import java.util.*;
8 public class Location {
9     public static void main(String[] args) {
10         HashMap<String, String> location = new HashMap<String, String>();
11         System.out.println("도시, 경도, 위도를 총 4회 입력하세요.");
12         Scanner scanner = new Scanner(System.in);
13         //4개의 input을 받습니다. 그리고 해쉬맵에 넣습니다.
14         for(int i = 0; i < 4; i++) {
15             String city, longitude, latitude;
16             city = scanner.next();
17             city = city.replace(", ", "");
18             longitude = scanner.next();
19             longitude = longitude.replace(", ", "");
20             latitude = scanner.next();
21             String point = longitude + latitude; //위도와 경도를 합쳐서 하나의 문자열로 만들어준다.
22             location.put(city, point);
23         }
24         //해쉬맵에 있는 모든 도시 출력.
25         System.out.println("해쉬맵에 있는 모든 도시를 출력합니다.");
26         Set<String> keys = location.keySet();
27         Iterator<String> iter = keys.iterator();
28         while(iter.hasNext()) {
29             String iter_city = iter.next();
30             String iter_point = location.get(iter_city);
31             System.out.println(iter_city + ", " + iter_point);
32         }
33         //해쉬맵에서 키를 검색에서 있다면 값을 출력, 없다면 없다고, 종료를 입력시 종료.
34         String city_search;
35         scanner.nextLine(); // scanner buffer clear
36         //Scanner search = new Scanner(System.in);
37         do{
38             System.out.println("도시를 검색합니다.");
39             city_search = scanner.nextLine();
40             if(city_search.equals("종료")) {
41                 System.out.println("검색을 종료합니다.");
42                 break;
43             }
44             String point_search = location.get(city_search);
45             //찾는 도시가 없을 경우
46             if(point_search == null)
47                 System.out.println(city_search + " 없습니다.");
48             //찾는 도시가 있을 경우 -> 출력!
49             else {
50                 System.out.println(city_search + " " + location.get(city_search).replace(", ", ""));
51             }
52         }while(true);
53     }
54 }
```

Location class 실행결과



```
<terminated> Location [Java Application] /Library/Ja
도시, 경도, 위도를 총 4회 입력하세요.
서울, 37, 126
LA, 34, -118
파리, 2, 48
시드니, 151, -33
해시맵에 있는 모든 도시를 출력합니다.
서울, 37, 126
LA, 34, -118
파리, 2, 48
시드니, 151, -33
도시를 검색합니다.
부산
부산 없습니다.
도시를 검색합니다.
파리
파리 2 48
도시를 검색합니다.
종료
검색을 종료합니다.
```

감사합니다.