

자바 기초 문법

클래스 String

System.out.printf()

```
public class Test {  
    public static void main(String[] args) {  
        String    id="P-001";  
        int        age=35;  
        double     height=175.843;  
        char       gender='남';  
        boolean    foreignerYN=false;  
  
        System.out.printf("아이디=%s\n", id); // %s => 대응하는 값을 문자열로 교체  
        System.out.printf("나이=%d(세)\n", age); // %d => 대응하는 정수를 십진수로 교체  
        System.out.printf("신장=%f(cm)\n", height); // %f => 대응하는 실수를 십진수로 교체  
        System.out.printf("성별=%c\n", gender); // %c => 대응하는 문자로 교체  
        System.out.printf("외국인여부=%b\n", foreignerYN); // %b => 대응하는 불리언 값으로 교체  
  
        System.out.printf("아이디=%s, 나이=%d(세), 신장=%f(cm), 성별=%c, 외국인여부=%b\n", id, age, height, gender, foreignerYN);  
    }  
}
```

실행결과

```
아이디=P-001  
나이=35 (세)  
신장=175.843000 (cm)  
성별=남  
외국인여부=false  
아이디=P-001, 나이=35 (세), 신장=175.843000 (cm), 성별=남, 외국인여부=false
```

System.out.printf()

```
public class Test {
    public static void main(String[] args) {
        String    id="P-001";
        int        age=35;
        double     height=175.843;
        char       gender='남';
        boolean    foreignerYN=false;

        System.out.printf("나이는%d(세)\n", age);
        System.out.printf("나이는%05d(세)\n", age); // 5개 문자 영역 내 오른쪽 정렬, 앞쪽 빈 여백 0 채움
        System.out.printf("나이는%5d(세)\n", age); // 5개 문자 영역 내 오른쪽 정렬
        System.out.printf("나이는%-5d(세)\n", age); // 5개 문자 영역 내 왼쪽 정렬
        System.out.printf("신장은%f(cm)\n", height);
        System.out.printf("신장은%.2f(cm)\n", height); // %.2f => 소수점 2째자리까지
        System.out.printf("신장은%08.2f(cm)\n", height); // 8개 문자 영역 내 오른쪽 정렬, 앞쪽 빈 여백 0 채움
        System.out.printf("신장은%8.2f(cm)\n", height); // 8개 문자 영역 내 오른쪽 정렬
        System.out.printf("신장은%-8.2f(cm)\n", height); // 8개 문자 영역 내 왼쪽 정렬

        System.out.printf("십진수 %d의 16진수 표현은 %x\n", age, age); // 35 = 16*2+1*3

        // System.out.printf("신장은%d(cm)\n", height); // 오류 발생
        // System.out.printf("나이는%f(세)\n", age); // 오류 발생
        System.out.printf("%s %s %s %s %s", id, age, height, gender, foreignerYN);
    }
}
```

실행결과

```
나이는35 (세)
나이는00035 (세)
나이는   35 (세)
나이는35    (세)
신장은175.843000 (cm)
신장은175.84 (cm)
신장은00175.84 (cm)
신장은  175.84 (cm)
신장은175.84   (cm)
십진수 35의 16진수 표현은 23
P-001 35 175.843 남 false
```

문자열 클래스 String: 문자열 객체 생성

```
public class Test {  
    public static void main(String[] args) {  
        String s1;  
        s1 = "경성대학교";  
  
        String s2 = "경성대학교";  
  
        String s3;  
        s3 = new String("경성대학교");  
  
        String s4 = new String("경성대학교");  
  
        System.out.println(s1); // 경성대학교  
        System.out.println(s2); // 경성대학교  
        System.out.println(s3); // 경성대학교  
        System.out.println(s4); // 경성대학교  
    }  
}
```

문자열 클래스 String: 문자 vs. 문자열

```
public class Test {  
    public static void main(String[] args) {  
        char    c = 'A';           // 문자  
        String  s = "A";           // 문자열  
        char    charArray[] = { 'A' }; // 문자 배열  
        String  stringArray[] = { "A" }; // 문자열 배열  
        System.out.println( c );     // 문자 'A' 출력  
        System.out.println( s );     // 문자열 "A" 출력  
        System.out.println( charArray[0] ); // 문자 'A' 출력  
        System.out.println( stringArray[0] ); // 문자열 "A" 출력  
    }  
}
```

문자열 클래스 String: 문자열 vs. 문자배열

```
public class Test {  
    public static void main(String[] args) {  
        char c[]={'경','성','대','학','교'};    // 문자배열  
        for (int i = 0; i < c.length; i++) {  
            System.out.print(c[i]);  
        }  
        String s="경성대학교";                // 문자열  
        System.out.println(s);  
  
        System.out.println( c[0] );            // 문자배열의 0번째 문자 출력  
        // System.out.println( s[0] );          // 오류발생  
    }  
}
```

문자열 변수에 대한 배열 인덱스 접근 불가

문자열 클래스 String: 문자열 vs. 문자배열

```
public class Test {  
    public static void main(String[] args) {  
        char c[]={'경','성','대','학','교'};    // 문자배열  
        for (int i = 0; i < c.length; i++) {  
            System.out.print(c[i]);  
        }  
        String s="경성대학교";                // 문자열  
        System.out.println(s);  
  
        System.out.println( c[0] );            // 문자배열의 0번째 문자 출력  
        // System.out.println( s[0] );          // 오류  
    }  
}
```

s[0]

- 문자열 변수 s에 대해 배열 인덱스 방식의 접근 불가

문자열 클래스 String: toCharArray()

```
public class Test {  
    public static void main(String[] args) {  
        String s="경성대학교";  
  
        char c[] = s.toCharArray();    // 문자열 => 문자배열 변환  
        for (int i = 0; i < c.length; i++) {  
            System.out.println(c[i]);  
        }  
  
        String t = new String(c);    // 문자배열 => 문자열 변환  
        System.out.println(t);  
        System.out.println(s.equals(t));  
    }  
}
```

s.toCharArray()

- 문자열 s에 대응하는 문자배열을 반환하는 메소드 호출문

new String(c)

- 문자배열 c를 파라미터로 전달받아 대응하는 문자열 객체를 생성하여 반환하는 메소드 호출문

문자열 클래스 String: charAt()

```
public class Test {  
    public static void main(String[] args) {  
        String s="경성대학교";  
  
        char c = s.charAt(0);           // 문자열 s의 0번째 위치 문자 반환  
        System.out.println(c);  
  
        System.out.println(s.charAt(1)); // 문자열 s의 1번째 문자 출력  
        //System.out.println(s[1]);      // 오류  
        System.out.println(s.charAt(4)); // 문자열 s의 4번째 문자 출력  
        // System.out.println(s.charAt(5)); // 오류  
    }  
}
```

s.charAt(0)

- 문자열 s의 0번째 문자를 반환하도록 요청하는 메소드 호출문

문자열 클래스 String: length()

```
public class Test {  
    public static void main(String[] args) {  
        char c[]={'경','성','대','학','교'};  
        System.out.println( c.length );           // c.length => 문자배열 c의 크기  
        System.out.println( c[c.length-1] );       // 문자배열 c의 마지막 위치 문자 출력  
  
        String s="경성대학교";  
        int len = s.length() ;                     // s.length() => 문자열 s의 길이 (문자 개수)  
        System.out.println(len);  
        System.out.println( s.charAt(s.length()-1) ); // 문자열 s의 마지막 위치 문자 출력  
    }  
}
```

c.length

- 문자배열 c의 크기(int)

s.length()

- 문자열 s의 길이(문자열 내 총 문자 개수)를 정수(int)로 반환하는 메소드 호출문

문자열 클래스 String: equals()

```
public class Test {  
    public static void main(String[] args) {  
        int m = 10, n = 20; // 기본 자료형 변수 m, n에는 각각 정수 10, 20이 저장됨  
  
        if( m == n ) System.out.println("m과 n에 저장된 값 일치");  
        else System.out.println("m과 n에 저장된 값 불일치");  
  
        String x = "ABC"; // 참조 자료형 변수 x에는 "ABC" 문자열 객체에 대한 참조 값(메모리 주소)이 저장됨  
        String y = "A";  
        y=y+"BC";  
  
        System.out.println(x); // x가 참조하는 객체의 데이터를 문자열로 출력  
        System.out.println(y); // y가 참조하는 객체의 데이터를 문자열로 출력  
  
        if(x==y) System.out.println("참조 값 일치");  
        else System.out.println("참조 값 불일치");  
  
        if( x.equals(y) ) System.out.println("참조 객체 내 데이터 일치");  
        else System.out.println("참조 객체 내 데이터 불일치");  
    }  
}
```

실행결과

m과 n에 저장된 값 불일치
ABC
ABC
참조 값 불일치
참조 객체 내 데이터 일치

m == n

- 기본 자료형 변수에 저장된 값 비교
- ## x.equals(y)
- 참조변수가 참조하는 데이터 비교
 - x가 참조하는 객체 데이터와 y가 참조하는 객체 데이터가 일치하는 경우 불리언 true를 그렇지 않은 경우 false를 반환하는 메소드 호출문

문자열 클래스 String: replace()

```
public class Test {  
    public static void main(String[] args) {  
        String s = "경성대학교,경성대학교";  
  
        String t = s.replace("대", "고등");  
        System.out.println(s);           // 원 문자열 s는 변경 없음 => 경성대학교,경성대학교  
        System.out.println(t);           // 실행결과 => 경성고등학교,경성고등학교  
  
        s = s.replace("대", "고등");  
        System.out.println(s);           // 실행결과 => 경성고등학교,경성고등학교  
    }  
}
```

s.replace("대", "고등")

- 문자열 s 내 부분문자열 "대"를 "고등"으로 변경(교체) 후 새로운 문자열 객체 반환하는 메소드 호출문
- 원 문자열 s는 변경 없음

s = s.replace("대", "고등")

- 원 문자열 변경 위해 변수 s에 변경된 문자열 객체의 참조 값을 대입 필요

문자열 클래스 String: toUpperCase(), toLowerCase()

```
public class Test {  
    public static void main(String[] args) {  
        String s="Korea";  
  
        String t = s.toUpperCase(); // 문자열 s를 대문자로 변환한 새로운 문자열 객체를 반환  
        String v = s.toLowerCase(); // 문자열 s를 소문자로 변환한 새로운 문자열 객체를 반환  
        System.out.println(s);      // 원 문자열 s는 변경 없음: Korea  
        System.out.println(t);      // 실행결과: KOREA  
        System.out.println(v);      // 실행결과: korea  
    }  
}
```

s.toUpperCase()

- 문자열 s를 대문자로 변환한 새로운 문자열 객체를 반환하는 메소드 호출문
- 원 문자열 s는 변경 없음

s.toLowerCase()

- 문자열 s를 소문자로 변환한 새로운 문자열 객체를 반환하는 메소드 호출문
- 원 문자열 s는 변경 없음

문자열 클래스 String: concat()

```
public class Test {  
    public static void main(String[] args) {  
        String s="경성대학교";  
  
        String t = s.concat("소프트웨어학과");  
  
        System.out.println(s); // 원 문자열 s는 변경 없음: 경성대학교  
        System.out.println(t); // 실행결과: 경성대학교소프트웨어학과  
    }  
}
```

s.concat("소프트웨어학과")

- 문자열 s의 마지막에 파라미터로 전달된 문자열 "소프트웨어학과"를 연결하여 얻어진 새로운 새로운 문자열 객체를 반환하는 메소드 호출문
- 원 문자열 s는 변경 없음

문자열 클래스 String: contains()

```
public class Test {  
    public static void main(String[] args) {  
        String s="경성대학교";  
  
        boolean flag = s.contains("대학");  
  
        System.out.println(flag);           // 실행결과: true  
        System.out.println( s.contains("고등") ); // 실행결과: false  
    }  
}
```

s.contains("대학")

- 문자열 s가 문자열 "대학"을 포함하는 경우 불리언 값 true를 반환하고 그렇지 않은 경우 불리언 값 false를 반환하는 메소드 호출문

문자열 클래스 String: startsWith(), endsWith()

```
public class Test {  
    public static void main(String[] args) {  
        String s="경성대학교";  
  
        boolean flag = s.endsWith("대학교");  
  
        System.out.println(flag); // 실행결과: true  
        System.out.println( s.endsWith("고등학교") ); // 실행결과: false  
  
        flag = s.startsWith("경성");  
        System.out.println(flag); // 실행결과: true  
        System.out.println( s.startsWith("한국") ); // 실행결과: false  
    }  
}
```

s.endsWith("대학교")

- 문자열 s가 문자열 "대학교"로 끝나는 경우 불리언 값 true를 반환하고 그렇지 않은 경우 불리언 값 false를 반환하는 메소드 호출문

s.startsWith("경성")

- 문자열 s가 문자열 "경성"으로 시작하는 경우 불리언 값 true를 반환하고 그렇지 않은 경우 불리언 값 false를 반환하는 메소드 호출문

문자열 클래스 String: indexOf()

```
public class Test {
    public static void main(String[] args) {
        String s="경성대학교 경성고등학교 경성중학교";

        System.out.println( s.indexOf('경') );    // 0
        System.out.println( s.indexOf('성') );    // 1
        System.out.println( s.indexOf(' ') );    // 5
        System.out.println( s.indexOf('중') );    // 15
        System.out.println( s.indexOf('한') );    // -1
        System.out.println( s.indexOf("경") );    // 0
        System.out.println( s.indexOf("대학교") ); // 2

        int index = -1;
        index = s.indexOf('학');
        System.out.println(index);    // 3
        index = s.indexOf('학', 4);
        System.out.println(index);    // 10
        index = s.indexOf('학', 11);
        System.out.println(index);    // 16
        index = s.indexOf('학', 17);
        System.out.println(index);    // -1
    }
}
```

s.indexOf('경')

- 문자열 s 내 0번째 위치 문자부터 탐색하여 문자 '경'의 최초 발견 위치 인덱스를 정수(int)로 반환하는 메소드
호출문으로, 미발견 시 -1 반환

s.indexOf("대학교")

- 문자열 s 내 0번째 위치 문자부터 탐색하여 문자열 "대학교"의 최초 발견 시작 위치 인덱스를 정수(int)로 반환하는 메소드
호출문으로, 미발견 시 -1 반환

s.indexOf('학', 4)

- 문자열 s 내 4번째 위치 문자부터 탐색하여 문자 '학'이 발견되는 위치 인덱스를 정수(int)로 반환하는 메소드
호출문으로, 미발견 시 -1 반환

문자열 클래스 String: lastIndexOf()

```
public class Test {  
    public static void main(String[] args) {  
        String s="경성대학교 소프트웨어학과";  
  
        System.out.println( s.lastIndexOf("소프트") ); // 6  
  
        int index = s.lastIndexOf("학");  
  
        System.out.println(index); // 11  
        System.out.println( s.lastIndexOf('학', 10) ); // 3  
        System.out.println( s.lastIndexOf('학', 2) ); // -1  
    }  
}
```

s.lastIndexOf("소프트")

- 문자열 s 내 마지막 위치 문자부터 역으로 탐색하여 문자열 "소프트"의 최초 발견 시작 위치 인덱스를 정수(int)로 반환하는 메소드 호출문으로, 미발견 시 -1 반환

s.lastIndexOf('학', 10)

- 문자열 s 내 10번째 위치 문자부터 역으로 탐색하여 문자 '학'이 발견되는 위치 인덱스를 정수(int)로 반환하는 메소드 호출문으로, 미발견 시 -1 반환

문자열 클래스 String: isEmpty(), "", null

```
public class Test {  
    public static void main(String[] args) {  
        String s = ""; // 변수 s에 길이 0 문자열 객체 참조 값 대입  
  
        boolean flag = s.isEmpty();  
  
        System.out.println(flag); // true  
        System.out.println(s.length()); // 0  
  
        String t = "경성대학교";  
        System.out.println(t.isEmpty()); // false  
        System.out.println(t.length()); // 5  
  
        String v = null; // 변수 v의 참조 값 없음  
        System.out.println(v); // null  
        // System.out.println(v.length()); // 오류 발생: java.lang.NullPointerException  
    }  
}
```

s.isEmpty()

- 문자열 s 내 문자 개수가 0인 경우
불리언 값 true를 반환하고 그렇지
않은 경우 불리언 값 false를
반환하는 메소드 호출문

문자열 클래스 String: substring()

```
public class Test {  
    public static void main(String[] args) {  
        String s="경성대학교";  
  
        String t = s.substring(2, 4);  
        System.out.println(t);  
  
        String v = s.substring(2);  
        System.out.println(v);  
    }  
}
```

s.substring(2, 4)

- 문자열 s 내 2번 위치 문자부터 4번 위치 직전 문자까지의 부분문자열에 해당하는 문자열 반환하는 메소드 호출문

s.substring(2)

- 문자열 s의 2번 위치 문자부터 마지막 위치 문자까지의 부분문자열에 해당하는 문자열 반환하는 메소드 호출문

문자열 클래스 String: trim()

```
public class Test {  
    public static void main(String[] args) {  
        String s = "\n  경성대학교\n\t ";  
  
        s.trim();  
        System.out.println(s);  
  
        s = s.trim();  
        System.out.println(s);  
    }  
}
```

실행결과

경성대학교

경성대학교

s.trim()

- 문자열 s의 앞부분 및 뒷부분 연속하는 공백 문자들을 제거한 문자열 객체를 반환하는 메소드 호출문
- 원 문자열 s는 변경 없음

문자열 클래스 String: split()

```
public class Test {  
    public static void main(String[] args) {  
        String s = "2019-03-12";  
  
        String t[] = s.split("-");  
  
        String year = t[0];    // "2012"  
        String month = t[1];   // "03"  
        String day = t[2];     // "12"  
        System.out.println(year+"년 "+month+"월 "+day+"일");  
    }  
}
```

s.split("-")

- 대쉬문자(-)를 분리자로 사용하여 분리되는 문자열 s의 부분문자열들의 배열을 반환하는 메소드 호출문
- split()의 파라미터로 전달되는 문자열은 정규표현식(regular expression)이다
- 참조: https://ko.wikipedia.org/wiki/정규_표현식

문자열 클래스 String: split()

```
public class Test {  
    public static void main(String[] args) {  
        String s = "서울시청부산시청대전시청대구시청광주시청";  
        String v[] = s.split("시청");  
        for (int i = 0; i < v.length; i++) {  
            System.out.println(v[i]);  
        }  
    }  
}
```

실행결과

서울
부산
대전
대구
광주

s.split("시청")

- 문자열 "시청"을 분리자로 사용하여 분리되는 문자열 s의 부분문자열들의 배열을 반환하는 메소드 호출문
- split()의 파라미터로 전달되는 문자열은 정규표현식(regular expression)이다
- 참조: https://ko.wikipedia.org/wiki/정규_표현식

문자열 클래스 String: split()

```
public class Test {  
    public static void main(String[] args) {  
        String s = "2019-12-25:03_12_59"; // 2019년11월12일 3시11분59초  
        String t[] = s.split("[-:_]");  
        for (int i = 0; i < t.length; i++) {  
            System.out.println(t[i]);  
        }  
    }  
}
```

실행결과

```
2019  
12  
25  
03  
12  
59
```

s.split("[-:_]")

- 정규표현식에서 []은 [] 내에 포함된 문자 중 하나를 의미한다
- s.split("[-:_]");은 [] 내에 포함된 세 개 문자인 대쉬(-), 콜론(:), 밑줄(_) 중 하나를 분리자로 사용하여 분리되는 문자열 s의 부분문자열들의 배열을 반환한다
- 위 코드의 s.split("[-:_]")을 s.split("12")로 변경하여 실행해 보고 split()의 사용법을 익히시오
- 참조: https://ko.wikipedia.org/wiki/정규_표현식

문자열 클래스 String: split()

```
public class Test {  
    public static void main(String[] args) {  
        String s = "2019 \r\n\t 03 \t\t\t \n\n\n 12";  
  
        String t[] = s.split("[ \\t\r\n]+");  
        for (int i = 0; i < t.length; i++) {  
            System.out.println(t[i]);  
        }  
    }  
}
```

실행결과

```
2019  
03  
12
```

s.split("[\\t\r\n]+")

- 정규표현식에서 +는 +이전 요소의 1회 이상 출현을 의미한다
- 아래 코드의 [\\t\r\n]+은 공백, 탭(\t), \r, \n 문자 중 하나가 1회 이상 연속 출현한 문자열을 분리자로 사용하여 문자열 s를 분리한다는 의미이다
- 위 코드의 s.split("[\\t\r\n]+")를 s.split("\\s+")로 대체하여 실행해 보시오
- 참조: https://ko.wikipedia.org/wiki/정규_표현식

문자열 클래스 String: split()

```
public class Test {  
    public static void main(String[] args) {  
        String s = "2019 \r\n\t 03 \t\t\t \n\n\n 12";  
        String t[] = s.split("[^0123456789]+");  
        for (int i = 0; i < t.length; i++) {  
            System.out.println(t[i]);  
        }  
    }  
}
```

실행결과

```
2019  
03  
12
```

s.split("[^0123456789]+")

- 위 코드의 [^0123456789]+에서 [^0123456789]은 10개 숫자 문자가 아닌 문자 하나를 의미한다
- s.split("[^0123456789]+")은 숫자 문자가 아닌 문자들의 나열을 분리자로 사용하여 문자열 s를 분리하므로, 문자열 s 내 숫자 문자열들의 배열을 반환한다
- s.split("[^0-9]+") 혹은 s.split("[^\\d]+")은 s.split("[^0123456789]+")와 같은 의미이다
- 참조: https://ko.wikipedia.org/wiki/정규_표현식

문자열 클래스 String: replaceAll()

```
public class Test {  
    public static void main(String[] args) {  
        String s = "이영희100\n김철수 95\n총점 195";  
        s = s.replaceAll("[^0-9]+", "");  
        System.out.println(s); // 실행결과: 10095195  
    }  
}
```

s.replaceAll("[^0-9]+", "")

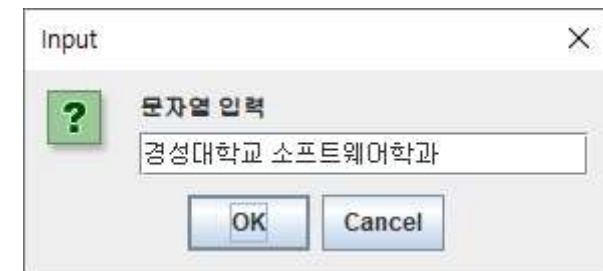
- s.replaceAll("[^0-9]+", "");에서 첫 파라미터는 정규표현식이며 두번째 파라미터는 문자열이다
- String의 replaceAll()은 첫 파라미터에 의해 매치되는 모든 문자열들을 두번째 파라미터로 교체한다
- s.replaceAll("[^0-9]+", "");은 문자열 s 내 길이 1이상의 모든 비슷자문자열들을 삭제한 새로운 문자열을 반환하며, 원 문자열 s는 변경 없음
- 참조: https://ko.wikipedia.org/wiki/정규_표현식

문자열 실습 A

- 다음은 입력창에 길이 1 이상의 문자열을 입력 후 OK버튼을 클릭하면 입력된 문자열의 **첫 문자**와 **마지막 문자**를 화면에 출력하는 코드와 그 실행 예시이다. 아래 코드를 완성하시오.

```
import javax.swing.JOptionPane;
public class Test {
    public static void main(String[] args) {
        String s = JOptionPane.showInputDialog("문자열 입력");

        char firstChar =      ;
        char lastChar =      ;
        System.out.println(firstChar);
        System.out.println(lastChar);
    }
}
```



실행결과

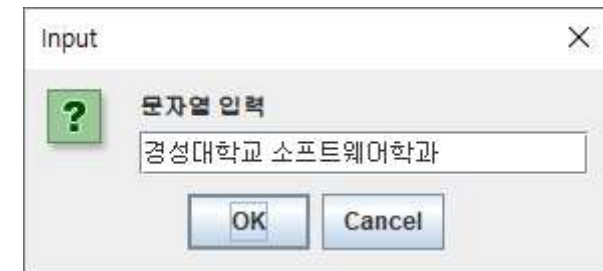
경
과

문자열 실습 B

- 다음은 입력창에 길이 3이상의 문자열을 입력 후 OK버튼을 클릭하면 입력된 문자열의 왼쪽 끝 세 문자들로 이루어진 문자열과 오른쪽 끝 세 문자들로 이루어진 문자열을 화면에 출력하는 코드와 그 실행 예시이다. 아래 코드를 완성하시오.

```
import javax.swing.JOptionPane;

public class Test {
    public static void main(String[] args) {
        String    s = JOptionPane.showInputDialog("문자열 입력");
        String    prefix =
        String    postfix =
        System.out.println(prefix);
        System.out.println(postfix);
    }
}
```

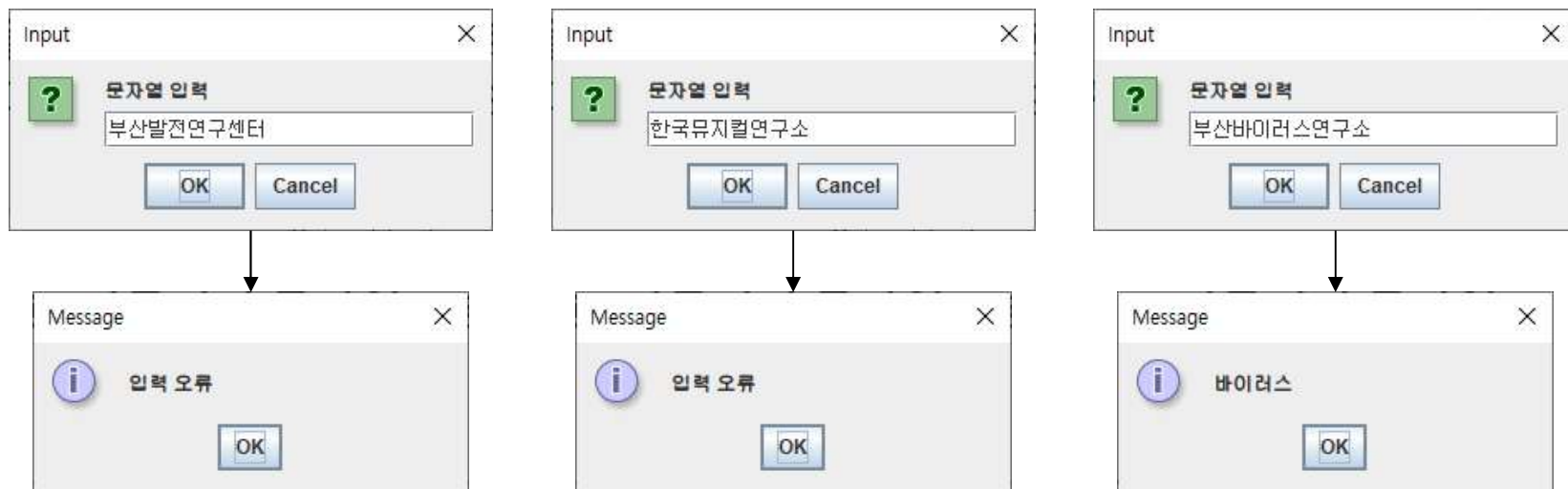


실행결과

경성대
어학과

문자열 실습 C

- 다음은 입력창에 “부산으로 시작하고 연구소로 끝나는” 문자열이 입력되면 부산과 연구소 사이의 문자열을 메시지창에 출력하고 그렇지 않은 경우 입력 오류를 메시지창에 출력하는 코드의 실행 예시이다. 이 코드를 완성하시오.
 - ✓ 예: 부산바이러스연구소 입력 시 → 바이러스 출력



문자열 실습 D

- 다음은 입력창에 길이 5이상의 문자열을 입력 후 OK버튼을 클릭하면 입력된 문자열 내에서 **문자열의 왼쪽부터 검사할 때 Seoul의 첫 출현 위치**를 화면에 출력하는 코드와 그 실행 예시이다. 아래 코드를 완성하시오.

```
import javax.swing.JOptionPane;

public class Test {
    public static void main(String[] args) {
        String    s = JOptionPane.showInputDialog("문자열 입력");
        int        pos =                ;
        System.out.println(pos);
    }
}
```



실행결과

0



실행결과

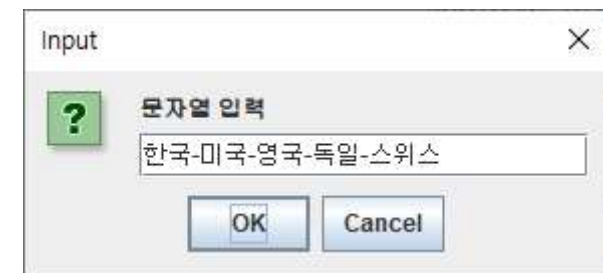
3

문자열 실습 E

- 다음은 입력창에 대쉬문자(-)로 분리된 2개 이상의 국가명을 입력 후 OK버튼을 클릭 하면 입력된 국가명 중 가장 왼쪽 국가명과 가장 오른쪽 국가명을 화면에 출력하는 코드와 그 실행 예시이다. 아래 코드를 완성하시오.

```
import javax.swing.JOptionPane;

public class Test {
    public static void main(String[] args) {
        String s = JOptionPane.showInputDialog("문자열 입력");
        String v[] = ;
        String firstCountry = ;
        String lastCountry = ;
        System.out.println(firstCountry);
        System.out.println(lastCountry);
    }
}
```

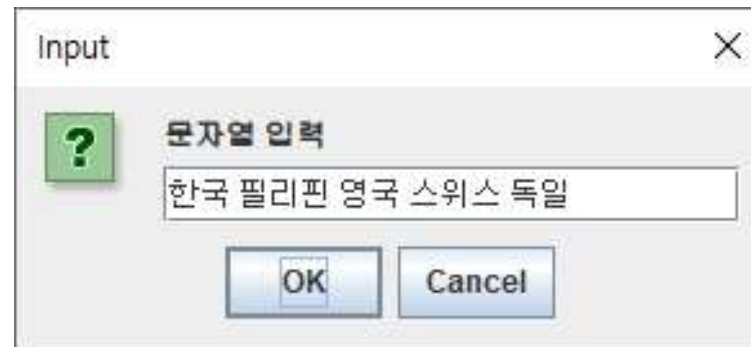


실행결과

한국
스위스

문자열 실습 F

- 다음은 입력창에 공백으로 분리된 2개 이상의 국가명을 입력 후 OK버튼을 클릭하면 입력된 국가명 중 3음절 이상의 국가명을 화면에 출력하는 코드의 실행 예시이다. 이 코드를 완성하시오.



실행결과

필리핀
스위스

문자열 실습 G

- 다음은 입력창에 대쉬문자(-)로 분리된 2개 이상의 국가명을 입력 후 OK버튼을 클릭하면 입력된 문자열 내 대쉬문자(-)를 콤마문자(,)로 변경하여 화면에 출력하는 코드의 실행 예시이다. 이 코드를 완성하시오.

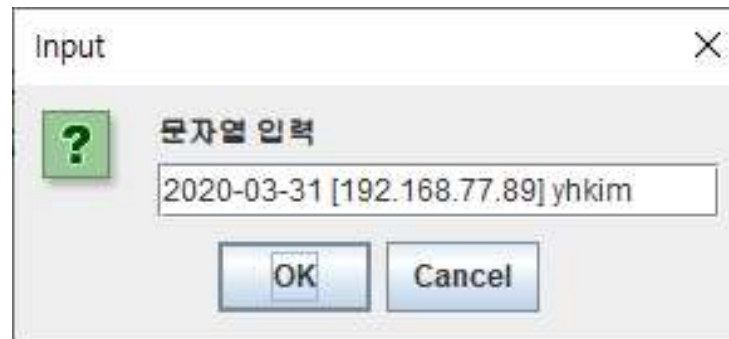


실행결과

한국,필리핀,영국,스위스,독일

문자열 실습 H

- 다음은 입력창에 로그문자열 입력 후 OK버튼을 클릭하면 입력된 문자열 내 IP주소를 화면에 출력하는 코드의 실행 예시이다. 이 코드를 완성하시오.
- 로그문자열 형식은 **연-월-일 [IP주소] 아이디**이며 IP주소는 두 괄호문자 [와]로 감싸져 있다. 로그문자열에서 괄호문자 [와]는 각각 IP주소 앞과 뒤에만 출현한다고 가정하시오
- 로그문자열 예시는 다음과 같다
 - ✓ 2020-03-31 [192.168.77.89] yhkim
 - ✓ 1987-12-25 [192.168.241.123] cspark

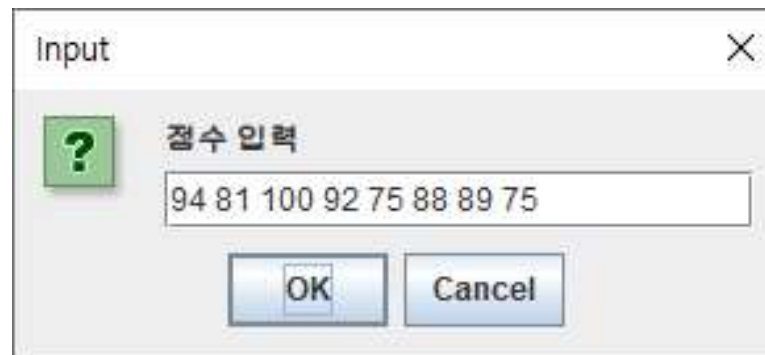


실행결과

192.168.77.89

문자열 실습 I

- 다음은 입력창에 공백으로 분리된 1개 이상의 점수(들)을 입력 후 OK버튼을 클릭하면 입력된 점수(들)의 평균을 소수점 둘째 자리까지 출력하는 코드의 실행 예시이다. 이 코드를 완성하시오.



Input

점수 입력

94 81 100 92 75 88 89 75

OK Cancel

실행결과

평균 = 86.75

문자열 실습 J

- 다음은 입력창에 문장을 입력하면 목적격조사 **을** 혹은 **를**이 부착된 어절 다음 어절을 출력하는 코드의 실행 예시이다. 이 코드를 완성하시오



실행결과

위해
탔다.

문자열 실습 K

- 다음은 입력창에 문장을 입력하면 입력된 문장의 어절들을 역순 출력하는 코드의 실행 예시이다. 이 코드를 완성하시오



실행결과

탔다. 비행기를 위해 방문을 한국 그들은

문자열 실습 1

- 다음은 입력창에 문장을 입력하면 입력된 문장의 문자들을 역순 출력하는 코드의 실행 예시이다. 이 코드를 완성하시오



실행결과

.다탸 를기행비 해위 을문방 국한 은들그

문자열 실습 M

- 다음은 입력창에 문장을 입력하면 입력된 문장의 각 어절을 역순 출력하는 코드의 실행 예시이다. 이 코드를 완성하시오

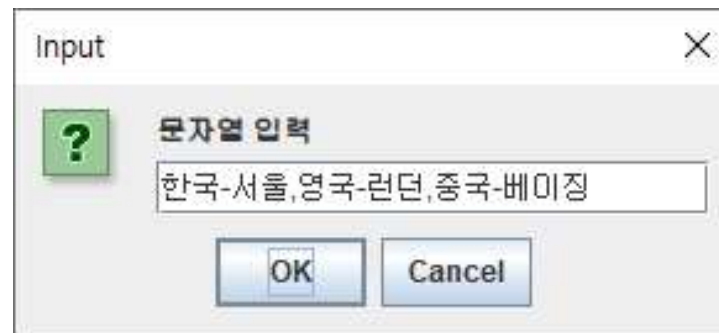


실행결과

은들그 국한 을문방 해위 를기행비 .다탸

문자열 실습 2

- 다음은 입력창에 콤마(,)로 분리된 국가명-수도명을 입력 후 OK버튼을 클릭하면 입력된 문자열에서 국가명과 수도명을 분리하여 출력하는 코드의 실행 예시이다. 이 코드를 완성하시오.
- 입력 문자열 예시(국가명과 수도명은 대쉬문자(-)로 분리되어 있으며, 국가명-수도명 사이에는 콤마 문자로 분리되어 있음)
 - ✓ 한국-서울,영국-런던,중국-베이징
 - ✓ 인도네시아-자카르타,필리핀-마닐라




실행결과

한국,영국,중국
서울,런던,베이징

References

 <http://docs.oracle.com/javase/7/docs/api/>

 <https://docs.oracle.com/javase/tutorial/java/>

 김윤명. (2008). 뇌를 자극하는 Java 프로그래밍. 한빛미디어.

 남궁성. 자바의 정석. 도우출판.

 황기태, 김효수 (2015). 명품 Java Programming. 생능출판사.