



STRING

String

❖ Java.lang.String

- String 클래스에는 문자열과 관련된 작업을 할 때 유용하게 사용할 수 있는 다양한 메소드가 포함되어 있다.

예제

```
public static void main(String[] args) {  
    Scanner input = new Scanner(System.in);  
    String data1;  
    String data2 = new String();  
  
    System.out.print("문자열 입력 : ");  
    data1 = input.next();  
    System.out.println("data1 : " + data1);  
    System.out.print("문자열 입력 : ");  
    data2 = input.next();  
    System.out.println("data2 : " + data2);  
}
```

예제

```
public static void main(String[] args) {  
    String data1 = "Java is Easy. 그리고 programming 할만하다.";  
    String data2;  
  
    System.out.println("출력 : "+ data1);  
    data2 = data1.toUpperCase();  
    System.out.println("변환 후 : " + data2);  
  
    data2 = data1.toLowerCase();  
    System.out.println("변환 후 : " + data2);  
    data1.toLowerCase();  
    System.out.println("출력 : " + data1);  
}
```

예제

```
public static void main(String[] args) {  
    String str = "abcdefg";  
  
    System.out.println(str.length());  
    System.out.println(str.charAt(1));  
    System.out.println(str.charAt(0) == 'a');  
    System.out.println(str.charAt(0) == 'c');  
    System.out.println(str.charAt(0)-32);  
    System.out.println((char)(str.charAt(0)-32));  
  
    String addStr;  
    addStr="a";  
    System.out.println(addStr);  
    addStr+="b";  
    System.out.println(addStr);  
}
```

예제

```
public static void main(String[] args) {  
    String str = new String(" Have a nice day ");  
  
    String result = str.trim();  
    System.out.println("--"+str+"--");  
    System.out.println("--"+result+"--");  
  
    String[] split = result.split(" ");  
    for(String p: split)  
        System.out.println(p);  
  
    String replace = result.replace(" ", "^");  
    System.out.println("replace : "+ replace);  
}
```



컬렉션 프레임워크 이해



컬렉션 프레임워크

❖ 컬렉션

- 객체의 수집 및 저장

❖ 컬렉션 프레임워크(Collection Framework)

- 객체들을 효율적으로 추가, 삭제, 검색할 수 있도록 제동되는 라이브러리

인터페이스 분류		특징	구현 클래스
Collection	List 계열	- 순서를 유지하고 저장 - 중복 저장 가능	ArrayList, Vector, LinkedList
	Set 계열	- 순서를 유지하지 않고 저장 - 중복 저장 안됨	HashSet, TreeSet
Map 계열		- 키와 값의 쌍으로 저장 - 키는 중복 저장 안됨	HashMap, Hashtable, TreeMap, Properties

List 컬렉션

ArrayList

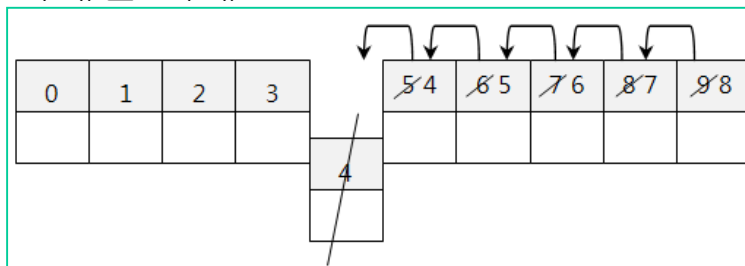
❖ 저장용량(Capacity)

- 초기용량은 10으로 고정되며 별도 지정 가능하고 초기 용량 초과 시 자동 확장



❖ 객체 제거

- 삭제된 객체로 shift



특징 및 종류

❖ 특징

- 인덱스로 관리
- 중복 객체 허용

❖ 종류

- Vector
- ArrayList
- LinkedList

기능	메소드	설명
객체 추가	<code>boolean add(E e)</code>	주어진 객체를 맨끝에 추가
	<code>void add(int index, E element)</code>	주어진 인덱스에 객체를 추가
	<code>set(int index, E element)</code>	주어진 인덱스에 저장된 객체를 주어진 객체로 바꿈
객체 검색	<code>boolean contains(Object o)</code>	주어진 객체가 저장되어 있는지 여부
	<code>E get(int index)</code>	주어진 인덱스에 저장된 객체를 리턴
	<code>isEmpty()</code>	컬렉션이 비어 있는지 조사
	<code>int size()</code>	저장되어있는 전체 객체수를 리턴
객체 삭제	<code>void clear()</code>	저장된 모든 객체를 삭제
	<code>E remove(int index)</code>	주어진 인덱스에 저장된 객체를 삭제
	<code>boolean remove(Object o)</code>	주어진 객체를 삭제

ArrayList

```
import java.util.ArrayList;

public class ArrayListEx01 {
    public static void main(String[] args) {
        ArrayList list = new ArrayList();

        list.add(123);
        list.add(456);

        System.out.println(list.get(0));
        System.out.println(list.get(1));
        //System.out.println(list.get(2));
    }
}
```

ArrayList

```
import java.util.ArrayList;
public class ArrayListEx02 {
    public static void main(String[] args) {
        ArrayList list = new ArrayList();

        list.add("123");
        list.add("456");
        list.add("789");

        System.out.println(list.get(0));
        System.out.println(list.get(1));
        System.out.println(list.get(2));
        System.out.println("개수 : " + list.size());
    }
}
```

ArrayList

```
public static void main(String[] args) {  
    Scanner input = new Scanner(System.in);  
    ArrayList arr = new ArrayList();  
  
    String addNum, printNum;  
    System.out.print("만들 공간의 개수 입력 : ");  
    int num = input.nextInt();  
    for(int i = 0; i < num; i++) {  
        System.out.print("데이터 입력 : ");  
        addNum = input.next();  
        arr.add(addNum);  
    }  
    System.out.println("--- 저장된 값 출력!! ---");  
    for(int i = 0; i < arr.size(); i++) {  
        printNum = (String)arr.get(i);  
        System.out.println(printNum);  
    }  
}
```

ArrayList

```
public static void main(String[] args) {  
    ArrayList arr = new ArrayList();  
  
    arr.add("123");  
    arr.add("456");  
  
    System.out.println(arr.contains("123"));  
    System.out.println(arr.contains("12"));  
  
    String searchData = "123";  
    if (arr.contains(searchData)) {  
        System.out.println(searchData+"데이터가 존재합니다.");  
    }else  
        System.out.println(searchData+"데이터가 존재하지 않습니다.");  
}
```

ArrayList

```
public static void main(String[] args) {  
    ArrayList arrList = new ArrayList();  
  
    arrList.add("123");  
    arrList.add("456");  
  
    System.out.println("지우기 전 [0]: "+arrList.get(0));  
    System.out.println("지우기 전 [1]: "+arrList.get(1));  
    System.out.println("결과 값 : "+arrList.remove("123"));  
    System.out.println("지우기 후 [0]: "+arrList.get(0));  
}
```


ArrayList

```
public static void main(String[] args) {  
    ArrayList arrClear = new ArrayList();  
  
    arrClear.add("123");  
    arrClear.add("456");  
  
    System.out.println(arrClear.get(0));  
    System.out.println(arrClear.get(1));  
  
    arrClear.clear(); // 모두 삭제  
  
    System.out.println("삭제 후");  
    //System.out.println(arrClear.get(0));  
    //System.out.println(arrClear.get(1));  
}
```

ArrayList

```
public static void main(String[] args) {  
    ArrayList list = new ArrayList();  
    list.add("김상수");list.add("이변수");list.add("박참조");  
    list.add("조클래스"); list.add("지컬렉션");  
  
    System.out.println(list);  
    System.out.print("변경할 위치 : ");  
    Scanner input = new Scanner(System.in);  
    int index = input.nextInt();  
  
    System.out.print("변경 데이터 : ");  
    String element = input.next();  
    list.set(index, element);  
  
    System.out.println("===== 결과 =====");  
    for(int i = 0; i < list.size(); i++)  
        System.out.println(list.get(i));  
}
```

ArrayList

```
public static void main(String[] args) {  
    ArrayList list = new ArrayList();  
  
    list.add("김상수"); list.add("이변수"); list.add("박참조");  
    list.add("조클래스"); list.add("지컬렉션");  
  
    System.out.println(list);  
    System.out.print("찾는 값 입력 : ");  
    Scanner input = new Scanner(System.in);  
    String value = input.next();  
  
    System.out.println(value + " 위치 : " + list.indexOf(value));  
}
```

Set

특징 및 종류

❖ 특징

- 임의의 저장순서
- 중복 저장 불가

❖ 종류

- HashSet
- TreeSet

기능	메소드	설명
객체 추가	boolean add(E e)	주어진 객체를 저장, 객체가 성공적으로 저장되면 true 를 리턴하고 중복 객체면 false 를 리턴
	boolean contains(Object o)	주어진 객체가 저장되어 있는지 여부
객체 검색	isEmpty()	컬렉션이 비어 있는지 조사
	Iterator<E> iterator()	저장된 객체를 한번씩 가져오는 반복자 리턴
	int size()	저장되어있는 전체 객체수 리턴
객체 삭제	void clear()	저장된 모든 객체를 삭제
	boolean remove(Object o)	주어진 객체를 삭제

HashSet

```
import java.util.ArrayList;
import java.util.HashSet;
public class HashSetEx01 {
    public static void main(String[] args) {
        HashSet hs = new HashSet();
        hs.add("라면");    hs.add("김밥");
        hs.add("순대");    hs.add("김밥");
        System.out.println("HashSet : "+hs);

        ArrayList arr = new ArrayList();
        arr.add("라면");    arr.add("김밥");
        arr.add("순대");    arr.add("김밥");
        System.out.println("ArrayList : "+arr);
    }
}
```

HashSet

```
import java.util.HashSet;
public class HashSetEx02 {
    public static void main(String[] args) {
        HashSet hs = new HashSet();
        boolean bool;
        bool = hs.add("라면");    System.out.println(bool);
        bool = hs.add("고길동"); System.out.println(bool);
        bool = hs.add("현대인"); System.out.println(bool);
        bool = hs.add("고길동"); System.out.println(bool);
        System.out.println("HashSet : "+hs);
    }
}
```

HashSet

```
import java.util.HashSet;
public class HashSetEx03 {
    public static void main(String[] args) {
        HashSet hs = new HashSet();
        hs.add("라면");hs.add("고길동");
        hs.add("현대인");hs.add("고길동");
        System.out.println("HashSet : "+hs);

        System.out.println(hs.remove("고길동"));
        System.out.println("HashSet : "+hs);
    }
}
```


HashSet

```
import java.util.HashSet;
import java.util.Iterator;
public class HashSetEx04 {
    public static void main(String[] args) {
        HashSet hs = new HashSet();
        hs.add("1.대부");      hs.add("2.반지의 제왕");
        hs.add("3.겨울왕국");  hs.add("4.다크 나이트");
        hs.add("5.해리포터");  hs.add("6.공공의 적");

        Iterator it = hs.iterator();
        while( it.hasNext() ) {
            String name = (String)it.next();
            System.out.println(name);
        }
    }
}
```

HashSet

```
public static void main(String[] args) {  
    HashSet hs = new HashSet();  
    hs.add("1.대부"); hs.add("2.반지의 제왕"); hs.add("3.겨울왕국");  
    hs.add("4.다크 나이트"); hs.add("5.해리포터"); hs.add("6.공공의 적");  
    System.out.println(hs);  
  
    Scanner input = new Scanner(System.in);  
    System.out.print("삭제 할 책 : ");  
    String name = input.nextLine();  
  
    Iterator it = hs.iterator();  
    while(it.hasNext()) {  
        if( name.equals((String)it.next()) ) {  
            System.out.println("삭제완료");  
            it.remove();  
        }  
    }  
}
```

HashSet

```
public static void main(String[] args) {  
    HashSet hs = new HashSet();  
    hs.add("1.대부"); hs.add("2.반지의 제왕"); hs.add("3.겨울왕국");  
    hs.add("4.다크 나이트"); hs.add("5.해리포터"); hs.add("6.공공의 적");  
    System.out.println(hs);  
  
    Scanner input = new Scanner(System.in);  
    System.out.print("삭제 할 책 : ");  
    String name = input.nextLine();  
  
    Iterator it = hs.iterator();  
    while(it.hasNext()) {  
        if( name.equals((String)it.next()) ) {  
            System.out.println("삭제완료");  
            it.remove();  
        }  
    }  
}
```

Map

특징 및 종류

❖ 특징

- Key value로 구성
- Key는 중복 불가능 하나 value는 중복 가능함

❖ 종류

- Hashmap, treemap

기능	메소드	설명
객체 추가	<code>V put(K key, V value)</code>	주어진 키와 값을 추가, 저장되면 값을 리턴
객체 검색	<code>boolean containsKey(Object key)</code>	주어진 키가 있는지 여부
	<code>boolean containsValue(Object value)</code>	주어진 값이 있는지 여부
	<code>Set<Map.Entry<K,V>> entrySet()</code>	키와 값의 쌍으로 구성된 모든 Map.Entry 객체를 Set에 담아서 리턴
	<code>V get(Object key)</code>	주어진 키의 값을 리턴
	<code>boolean isEmpty()</code>	컬렉션이 비어있는지 여부
	<code>Set<K> keySet()</code>	모든 키를 Set 객체에 담아서 리턴
	<code>int size()</code>	저장된 키의 총 수를 리턴
	<code>Collection<V> values()</code>	저장된 모든 값 Collection에 담아서 리턴
객체 삭제	<code>void clear()</code>	모든 Map.Entry(키와 값)를 삭제
	<code>V remove(Object key)</code>	주어진 키와 일치하는 Map.Entry 삭제, 삭제가 되면 값을 리턴

HashMap

```
import java.util.HashMap;
public class HashMapEx01 {
    public static void main(String[] args){
        HashMap<Integer, String> hm = new HashMap<>();

        hm.put(0, "test");
        hm.put(1, "hello");

        System.out.println(hm.get(0));
        System.out.println(hm.get(1));
    }
}
```

HashMap

```
import java.util.HashMap;

public class HashMapEx02 {
    public static void main(String[] args) {
        HashMap<String, Integer> map = new HashMap<>();

        map.put("선풍기", 10);
        map.put("에어컨", 100);
        map.put("건조기", 100);

        System.out.println("선풍기 : "+map.get("선풍기") + "만원");
        System.out.println("에어컨 : "+map.get("에어컨") + "만원");
        System.out.println("건조기 : "+map.get("건조기") + "만원");
        System.out.println("없는값 : "+map.get("없는값") + "만원");
    }
}
```

HashMap

```
import java.util.HashMap;
public class HashMapEx03 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        HashMap<String, String> map = new HashMap<>();

        System.out.print("키 입력 : ");
        String key = input.next();

        System.out.print("값 입력 : ");
        String value = input.next();

        map.put(key, value);
        System.out.println(map);
    }
}
```


HashMap

```
import java.util.HashMap;
public class HashMapEx04 {
    public static void main(String[] args) {
        HashMap map = new HashMap();
        map.put("선풍기", 10); map.put("에어컨", 100);
        map.put("건조기", 100);
        System.out.println(map);

        Scanner input = new Scanner(System.in);
        System.out.print("키 입력 : "); String key = input.next();
        System.out.print("값 입력 : "); int value = input.nextInt();

        System.out.println(key + " 키 검색 결과 : " + map.containsKey(key));
        System.out.println(value + " 값 검색 결과 : " + map.containsValue(value));
    }
}
```

HashMap

```
public static void main(String[] args) {  
    HashMap map = new HashMap();  
  
    map.put("선풍기", "10만원");  
    map.put("에어컨", "100만원");  
    map.put("건조기", "200만원");  
  
    System.out.println(map.remove("에어컨"));  
    System.out.println("결과 : " + map.containsKey("에어컨"));  
    System.out.println(map);  
}
```

HashMap

```
public static void main(String[] args) {  
    HashMap map = new HashMap();  
  
    map.put("선풍기", 10);  
    map.put("에어컨", 100);  
    map.put("건조기", 200);  
  
    System.out.println("목록 : " + map.keySet());  
    System.out.println("목록 : " + map.values());  
}
```

HashMap

```
public static void main(String[] args) {  
    HashMap map = new HashMap();  
    map.put("선풍기", "10만원");    map.put("에어컨", "100만원");  
    map.put("건조기", "200만원");  
  
    Iterator it = map.keySet().iterator();  
    while(it.hasNext()) {  
        String key = (String)it.next();  
        System.out.println(key + " : " + map.get(key));  
    }  
  
    it = map.values().iterator();  
    while(it.hasNext()) {  
        String value = (String)it.next();  
        System.out.println(value);  
    }  
}
```

HashMap

```
public static void main(String[] args) {  
    HashMap map = new HashMap();  
  
    map.put("선풍기", 10);  
    map.put("에어컨", 100);  
    map.put("건조기", 200);  
  
    Scanner input = new Scanner(System.in);  
    System.out.print("입력 : ");  
    String key = input.next();  
  
    map.remove(key);  
  
    System.out.println("목록 : " + map.keySet());  
    System.out.println("목록 : " + map.values());  
}
```