

## *Lab Exercise #6*



# 제네릭과 컬렉션 프레임워크 실습

---

2018년도 2학기

컴퓨터프로그래밍2

김 영 국

충남대학교 컴퓨터공학과



# 목차

---

## ■ 실습

- 컬렉션 인터페이스 실습
- Collections 클래스 실습
- 로또 번호 생성 프로그램

## ■ 과제

- StringOneHot 클래스

## 실습6-1. 컬렉션 인터페이스 실습(1)

- 강의노트를 참고하여 컬렉션 인터페이스들을 사용해보자.
- ArrayList, LinkedList, HashSet, Queue, Map을 사용해본다.
- 코드는 아래와 같다.
  - ArrayList 실습 코드

```
public class ArrayListTest {  
    public static void main(String[] args) {  
        ArrayList<String> list = new ArrayList<>();  
  
        list.add("MILK");  
        list.add("BREAD");  
        list.add("BUTTER");  
        list.add(1, "APPLE");  
        list.set(2, "GRAPE");  
  
        for(int i = 0; i < list.size(); i++) {  
            System.out.println(list.get(i));  
        }  
    }  
}
```



## 실습6-1. 컬렉션 인터페이스 실습(2)

- LinkedList 실습 코드

```
public class LinkedListTest {  
    public static void main(String[] args) {  
        LinkedList<String> list = new LinkedList<>();  
  
        list.add("Zero");  
        list.add("One");  
        list.add("Two");  
        list.add("Three");  
        list.set(2, "이");  
        list.remove(3);  
  
        Iterator<String> e = list.iterator();  
  
        while(e.hasNext()) {  
            String s = e.next();  
            System.out.println(s);  
        }  
        while(e.hasNext()) {  
            String s = e.next();  
            System.out.println(s);  
        }  
    }  
}
```

# 실습6-1. 컬렉션 인터페이스 실습(3)

## ■ Set 실습 코드

```
public class SetTest {  
    public static void main(String[] args) {  
        HashSet<String> set = new HashSet<String>();  
  
        set.add("milk");  
        set.add("bread");  
        set.add("butter");  
        set.add("cheese");  
        set.add("ham");  
        set.add("ham");  
  
        System.out.println(set);  
    }  
}
```

## ■ Queue 실습 코드

```
public class QueueTest {  
    public static void main(String[] args) throws InterruptedException {  
        int time = 10;  
        Queue<Integer> queue = new LinkedList<>();  
        for(int i = time; i >= 0; i--)  
            queue.add(i);  
        while(!queue.isEmpty()) {  
            System.out.print(queue.remove() + " ");  
            Thread.sleep(1000);  
        }  
    }  
}
```

# 실습6-1. 컬렉션 인터페이스 실습(4)

## ■ Map 실습 코드

```
public class MapTest {
    public static void main(String[] args) {
        Map<String, Student> st = new HashMap<>();

        st.put("201800001", new Student(201800001, "유비"));
        st.put("201800002", new Student(201800002, "관우"));
        st.put("201800003", new Student(201800003, "장비"));

        //모든 항목을 출력
        System.out.println(st);

        //하나의 항목을 삭제
        st.remove("201800002");
        //하나의 항목을 대체
        st.put("201800003", new Student(201800003, "조자룡"));

        System.out.println(st.get("201800003"));

        for(Map.Entry<String, Student> s : st.entrySet()) {
            String key = s.getKey();
            Student value = s.getValue();
            System.out.println("key=" + key + ", value=" + value);
        }
    }
}
```

```
class Student {
    @SuppressWarnings("unused")
    private int number;
    private String name;

    public Student(int number, String name) {
        this.number = number;
        this.name = name;
    }

    public String toString() {
        return name;
    }
}
```

## 실습6-2. Collections 클래스 실습(1)

- Collections 클래스를 사용하여 다음 기능을 사용해보자.
  - 정렬(Sorting)
  - 섞기(Shuffling)
  - 탐색(Searching)
- 코드는 다음과 같다.
  - 정렬(Sorting) 코드

```
class Student implements Comparable<Student> {  
    private int number;  
    private String name;  
    public Student(int number, String name) {  
        this.number = number;  
        this.name = name;  
    }  
    @Override  
    public int compareTo(Student s) {  
        return number - s.number;  
    }  
    public String toString() {  
        return name;  
    }  
}
```

```
public class SortTest {  
    public static void main(String[] args) {  
        Student array[] = {  
            new Student(2018005, "김철수"),  
            new Student(2018002, "이철수"),  
            new Student(2018003, "박철수")  
        };  
        List<Student> list = Arrays.asList(array);  
        Collections.sort(list);  
        System.out.println(list);  
    }  
}
```

## 실습6-2. Collections 클래스 실습(2)

- 섞기(Shuffling) 코드

```
public class Shuffle {  
    public static void main(String[] args) {  
        List<Integer> list = new ArrayList<>();  
        for(int i = 1; i <=10; i++)  
            list.add(i);  
        Collections.shuffle(list);  
        System.out.println(list);  
    }  
}
```

- 탐색(Searching) 코드

```
public class Search {  
    public static void main(String[] args) {  
        int key = 50;  
        List<Integer> list = new ArrayList<Integer>();  
        for(int i = 0; i < 100; i++)  
            list.add(i);  
        int index = Collections.binarySearch(list, key);  
        System.out.println("탐색의 반환값 =" + index);  
    }  
}
```



## 실습6-3. 로또 번호 생성프로그램

- 로또 번호를 생성하는 프로그램을 작성하여 보자.
- 로또는 1부터 45까지의 숫자 중에서 6개를 선택한다.
- 코드는 아래와 같다.

- Lotto 클래스

```
class Lotto {  
    private LinkedList<Integer> lottoBalls;  
    private ArrayList<Integer> prizeNumber;  
  
    public Lotto() {  
        lottoBalls = new LinkedList<>();  
        prizeNumber = new ArrayList<>(6);  
        for (int i = 1; i <= 45; i++)  
            lottoBalls.add(i);  
        Collections.shuffle(lottoBalls);  
        for (int i = 0; i < 6; i++)  
            prizeNumber.add(lottoBalls.get(i));  
    }  
  
    public ArrayList<Integer> getPrize() {  
        Collections.sort(this.prizeNumber);  
        return this.prizeNumber;  
    }  
}
```

- Lotto 객체 생성 후 getPrize() 호출 결과

당첨 번호: [4, 19, 29, 38, 39, 41]



## 과제 6-1. StringOneHot 클래스

---

- One-hot encoding은 문자열의 개수를 벡터의 크기로 하고, 표현하고 싶은 단어의 인덱스에 1의 값을 부여하고, 다른 인덱스에는 0을 부여하는 단어의 벡터 표현 방식이다.
- 문자열을 입력하면 각각의 글자들을 One-hot 벡터로 만드는 클래스를 컬렉션 프레임워크를 활용하여 만들어 보자.

## 과제 6-1. StringOneHot 클래스 (계속)

- 해당 클래스의 테스트 결과는 아래와 같다.

```
public static void main(String[] args) {  
  
    String str = "Hello";  
    StringOneHot soh = new StringOneHot(str);  
    System.out.println("문자열 :" + soh.getString());  
    System.out.println("문자열 Set: " + soh.getStrSet());  
    System.out.println("문자열 Map: " + soh.getStrMap());  
    System.out.println("문자열 One-hot Vector :" + soh.getStrOneHotVec());  
  
    System.out.println("One-hot Vector로 나타낸 " + soh.getString());  
    for(int i = 0; i < str.length(); i++) {  
        System.out.println(soh.getStrOneHotVec().get(str.substring(i, i+1)));  
    }  
}  
  
문자열 :Hello  
문자열 Set: [e, H, l, o]  
문자열 Map: {0=e, 1=H, 2=l, 3=o}  
문자열 One-hot Vector :{e=[1, 0, 0, 0], H=[0, 1, 0, 0], l=[0, 0, 1, 0], o=[0, 0, 0, 1]}  
One-hot Vector로 나타낸 Hello  
[0, 1, 0, 0]  
[1, 0, 0, 0]  
[0, 0, 1, 0]  
[0, 0, 1, 0]  
[0, 0, 0, 1]
```



# 과제 제출 및 기한

---

- 제출 방법

- 사이버캠퍼스를 통하여 제출
- 소스코드를 제출

- 제출 기한

- 이번 주 토요일(10/27) 자정