

JAVA 웹 프로그래밍

실습 과제#17

학과명: ss전공

교수명: 김삼근 교수님

제출자: 이종수

학번: 2017250035

제출일: 2021년 6월 7일

1.

test를 yes로 했을 경우

```
1 package ch17_1;
2
3 import java.util.*;
4 import java.io.*;
5
6 public class Jukebox1 {
7     ArrayList<String> songList = new ArrayList<>();
8
9     public static void main(String[] args){
10         new Jukebox1().go();
11         System.out.println("by 2017250035 이종수");
12     }
13
14     public void go(){
15         getSongs();
16         System.out.println(songList);
17     }
18
19     void getSongs(){
20         try {
21             File file = new File( pathname: "/Users/leejongsu/IdeaProjects/Chapter17/src/ch17_1/SongList.txt");
22             BufferedReader reader = new BufferedReader(new FileReader(file));
23             String line = null;
24             while((line = reader.readLine()) != null){
25                 addSong(line);
26             }
27         } catch (Exception ex){ex.printStackTrace();}
28     }
29
30     void addSong(String linetoParse){
31         String[] tokens = linetoParse.split( regex: "/" );
32         songList.add(tokens[0]);
33     }
34 }
```

실행 결과

```
Run: Jukebox1 x
/Library/Java/JavaVirtualMachines/jdk-15.0.1.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA
[너무 보고싶어, Pink Moon, 호랑수월가, 호랑풍류가, 압구정 날라리, 반지하 인생, 거울아 거울아, 눈꽃, 살다가 한번쯤, 긴 생머리 그녀, 있길 없길]
by 2017250035 이종수
Process finished with exit code 0
```

String 타입의 ArrayList에 노래 제목을 저장합니다. 노래 제목 파일은 SongList.txt로 미리 만들었습니다. 이후 go() 메소드를 통해 파일로드를 시작한 다음 SongList ArrayList의 내용을 인쇄합니다. getSongs() 메소드에서는 파일을 읽고 while문을 돌면서 addSong() 메소드를 호출합니다. 파일을 읽기위해서 pathname에 절대 경로를 주었습니다. split("/") 메소드로 / 단위로 자르며, 노래 제목만 원하므로 SongList에 첫 번째 토큰만 추가합니다.(tokens[0]) 따라서 위와 같이 SongList에 title만 출력되는 것을 확인할 수 있습니다.

2.

```
1  package ch17_2;
2
3  import java.util.*;
4  import java.io.*;
5
6  public class Jukebox2 {
7      ArrayList<String> songList = new ArrayList<>();
8
9      public static void main(String[] args){
10         new Jukebox2().go();
11         System.out.println("by 2017250035 이종수");
12     }
13
14     public void go(){
15         getSongs();
16         System.out.println(songList);
17         Collections.sort(songList);
18         System.out.println(songList);
19     }
20
21     void getSongs(){
22         try {
23             File file = new File( pathname: "/Users/leejongsu/IdeaProjects/Chapter17/src/ch17_1/SongList.txt");
24             BufferedReader reader = new BufferedReader(new FileReader(file));
25             String line = null;
26             while((line = reader.readLine()) != null){
27                 addSong(line);
28             }
29         } catch (Exception ex){ex.printStackTrace();}
30     }
31
32     void addSong(String linetoParse){
33         String[] tokens = linetoParse.split( regex: "/" );
34         songList.add(tokens[0]);
35     }
36 }
```

실행 결과

```
Run: Jukebox2 x
/Library/Java/JavaVirtualMachines/jdk-15.0.1.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA
[너무 보고싶어, Pink Moon, 호랑수월가, 호랑풍류가, 압구정 날라리, 반지하 인생, 거울아 거울아, 눈꽃, 살다가 한번쯤, 긴 생머리 그녀, 있길 없길]
[Pink Moon, 거울아 거울아, 긴 생머리 그녀, 너무 보고싶어, 눈꽃, 반지하 인생, 살다가 한번쯤, 압구정 날라리, 있길 없길, 호랑수월가, 호랑풍류가]
by 2017250035 이종수
Process finished with exit code 0
```

1번 코드를 활용하고 Collections.sort() 메소드를 추가하여 title을 알파벳 순으로 정렬하려 출력하는 예제입니다. Collections.sort() 메소드는 문자열 목록을 알파벳순으로 정렬합니다. 정적 Collections.sort() 메소드를 호출 하고, 다음 목록을 다시 인쇄합니다. 따라서 두 번째 줄 출력은 알파벳순으로 정렬되어 출력된 것을 확인할 수 있습니다.

3. Jukebox3 class

```
1 package ch17_3;
2
3 import java.util.*;
4 import java.io.*;
5
6 public class Jukebox3 {
7     ArrayList<Song> songList = new ArrayList<Song>();
8
9     public static void main(String[] args){
10         new Jukebox3().go();
11         System.out.println("by 2017250035 이종수");
12     }
13
14     public void go(){
15         getSongs();
16         System.out.println(songList);
17         Collections.sort(songList);
18         System.out.println(songList);
19     }
20
21     void getSongs(){
22         try {
23             File file = new File( pathname: "/Users/leejongsu/IdeaProjects/Chapter17/src/ch17_3/SongListMore.txt");
24             BufferedReader reader = new BufferedReader(new FileReader(file));
25             String line = null;
26             while((line = reader.readLine()) != null){
27                 addSong(line);
28             }
29         } catch (Exception ex){ex.printStackTrace();}
30     }
31
32     void addSong(String linetoParse){
33         String[] tokens = linetoParse.split( regex: "/" );
34         Song nextSong = new Song(tokens[0], tokens[1], tokens[2], tokens[3]);
35         songList.add(nextSong);
36     }
37 }
```

Song class

```
1 package ch17_3;
2
3 public class Song implements Comparable<Song>{
4     String title;
5     String artist;
6     String rating;
7     String bpm;
8
9     public int compareTo(Song s) { return title.compareTo(s.getTitle()); }
10
11     Song(String t, String a, String r, String b){
12         title = t;
13         artist = a;
14         rating = r;
15         bpm = b;
16     }
17
18     public String getTitle() { return title; }
19
20     public String getArtist() { return artist; }
21
22     public String getRating() { return rating; }
23
24     public String getBpm() { return bpm; }
25
26     public String toString() { return title; }
27 }
```

2번 코드와 Generics를 활용하여 리스트에 단순히 String이 아니라 Song객체를 넣는 예제입니다. 먼저 SongList.txt에 2가지의 내용을 추가하여

SongListMore.txt를 만듭니다. 노래의 비트, Song class에 파일에 있는 노래 속성에 대한 4개의 인스턴스 변수를 선언하고 그에 해당하는 getter메소드를 생성합니다. 우리가 원하는 title을 보기 위해 toString()메소드를 override해 줍니다. title별로 정렬하기 위해 메소드에 전달된 Song의 제목과 compareTo() 메소드에서 호출한 Song의 제목을 비교하도록 compareTo() 메소드를 구현하고, 메소드를 실행하는 Song은 자신의

title과 메소드 파라미터의 제목을 비교하여 크고 작은지를 결정합니다. 정렬을 위해 Song 객체를 다른 Song 객체와 비교할 수 있습니다. sort() 메소드는 Song을 compareTo()로 보내 해당 Song이 메소드가 호출된 Song과 어떻게 비교되는지 확인합니다. 따라서 위와 같은 결과를 출력하는 것을 확인할 수 있습니다. **실행 결과**

```
Run: Jukebox3 x
/Library/Java/JavaVirtualMachines/jdk-15.0.1.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA.app/Contents/lib/idea-runtime.jar -Didea.config.path=/Applications/IntelliJ IDEA.app/Contents/lib/idea-runtime.jar -Didea.home.path=/Applications/IntelliJ IDEA.app/Contents/lib/idea-runtime.jar -Didea.platform.prefix=IntelliJ -jar /Applications/IntelliJ IDEA.app/Contents/lib/idea-runtime.jar
[너무 보고싶어, I hate u, I love u, Pink Moon, 호랑수월가, 호랑풍류가, 압구정 날라리, 반지하 인생, 거울아 거울아, 눈꽃, 살다가 한번쯤, 긴 생머리 그녀, 있기 없기]
[I hate u, I love u, Pink Moon, 거울아 거울아, 긴 생머리 그녀, 너무 보고싶어, 눈꽃, 반지하 인생, 살다가 한번쯤, 압구정 날라리, 있기 없기, 호랑수월가, 호랑풍류가]
by 2017250035 이종수
Process finished with exit code 0
```

4.

```

1 package ch17_4;
2 import java.util.ArrayList;
3 import java.util.Scanner;
4 public class GolfScores {
5     public static void main(String[] args){
6         ArrayList<Double> score = new ArrayList<>();
7         System.out.println("This program reads golf scores and shows");
8         System.out.println("how much each differs from the average");
9         System.out.println("Enter golf scores: ");
10        fillArrayList(score);
11        showDifference(score);
12        System.out.println("by 2017250035 이종수");
13    }
14    public static void fillArrayList(ArrayList<Double> a){
15        System.out.println("Enter a list of nonnegative numbers.");
16        System.out.println("Mark the end of list with a negative number.");
17        Scanner keyboard = new Scanner(System.in);
18        double next;
19        next = keyboard.nextDouble();
20        while (next >= 0){
21            a.add(next);
22            next = keyboard.nextDouble();
23        }
24    }
25    @ public static double computeAverage(ArrayList<Double> a){
26        double total = 0;
27        for (Double element : a) total = total + element;
28        int numberOfScores = a.size();
29        if (numberOfScores > 0){
30            return (total/numberOfScores);
31        }else {
32            System.out.println("ERROR: Trying to average 0 numbers.");
33            System.out.println("computeAverage returns 0.");
34            return 0;
35        }
36    }
37    public static void showDifference(ArrayList<Double> a){
38        double average = computeAverage(a);
39        System.out.println("Average of the " + a.size() + " scores = " + average);
40        System.out.println("The scores are: ");
41        for(Double element : a) System.out.println(element + " differs form average by " + (element - average));
42    }
43 }

```

실행 결과

```

Run: GolfScores x
/Library/Java/JavaVirtualMachines/jdk-15.0.1.jdk/
This program reads golf scores and shows
how much each differs from the average
Enter golf scores:
Enter a list of nonnegative numbers.
Mark the end of list with a negative number.
69 74 68 -1
Average of the 3 scores = 70.33333333333333
The scores are:
69.0 differs form average by -1.3333333333333286
74.0 differs form average by 3.6666666666666714
68.0 differs form average by -2.3333333333333286
by 2017250035 이종수

Process finished with exit code 0

```

사용자가 골프 점수를 입력하고, 입력한 점수를 평균하고 각각의 점수가 평균과 얼마나 떨어져있는지를 출력하는 예제입니다. fillArrayList() 메소드에서 점수 및 끝 번호를 입력하고, computeAverage() 함수에서 평균을 계산하고, showDifference에서 평균과 얼마나 차이가 나는지를 출력해줍니다.

따라서 위와 같은 출력이 나오는 것을 확인할 수 있습니다.

5. Pair class

```
1 package ch17_5;
2
3 public class Pair<T> {
4     private T first;
5     private T second;
6
7     public Pair(){
8         first = null;
9         second = null;
10    }
11
12    public Pair(T firstItem, T secondItem){
13        first = firstItem;
14        second = secondItem;
15    }
16
17    public void setFirst(T first) { this.first = first; }
18
19    public void setSecond(T second) { this.second = second; }
20
21    public T getFirst() { return first; }
22
23    public T getSecond() { return second; }
24
25    public String toString() { return ("first: " + first.toString() + "\n" + "second: " + second.toString()); }
26
27    public boolean equals(Object otherObject){
28        if (otherObject == null) return false;
29        else if (getClass() != otherObject.getClass()) return false;
30        else {
31            Pair<T> otherPair = (Pair<T>)otherObject;
32            return (first.equals(otherPair.first) && second.equals(otherPair.second));
33        }
34    }
35 }
```

GenericPairDemo class

```
1 package ch17_5;
2
3 import java.util.Scanner;
4
5 public class GenericPairDemo {
6     public static void main(String[] args){
7         Pair<String> secretPair = new Pair<String>("Happy", "Day");
8
9         Scanner keyboard = new Scanner(System.in);
10        System.out.println("Enter two words: ");
11        String word1 = keyboard.next();
12        String word2 = keyboard.next();
13        Pair<String> inputPair = new Pair<String>(word1, word2);
14        if (inputPair.equals(secretPair)){
15            System.out.println("You guessed the secret words");
16            System.out.println("in the correct order!");
17        }else {
18            System.out.println("You guessed incorrectly.");
19            System.out.println("You guessed");
20            System.out.println(inputPair);
21            System.out.println("The secret words are");
22            System.out.println(secretPair);
23        }
24        System.out.println("by 2017250035 이종수");
25    }
26 }
```

실행 결과

```
Run: GenericPairDemo x
/Library/Java/JavaVirtualMachines/j
Enter two words:
first second
You guessed incorrectly.
You guessed
first: first
second: second
The secret words are
first: Happy
second: Day
by 2017250035 이종수

Process finished with exit code 0
```

Generics를 활용하는 예제입니다. Pair 클래스에서 두개의 변수를 선언하고 디폴트 생성자, 2개의 파라미터가 있는 생성자를 만들어 주고 변수 2개의 getter, setter 메소드르 만들어 줍니다. toString을 override해줍니다. equals() 메소드를 만듭니다. 이 메소드는 otherObecit가 null 이거나 가져온 getClass()가 다를 경우 fasle를 리턴하고, first랑 otherObject.firt, second를 비교합니다. 이제 Drive 클래스를 만듭니다. 단어 두개를 Happy, Day로 설정하고, 사용자로부터 입력을 받은 뒤 두 단어를 비교해 같은지 다른지를 출력하고 사용자가 입력한 단어를 출력하고, 설정된 단어를 출력합니다 따라서 위와 같은 결과가 나오는 것을 확인할 수 있습니다.

6. TwoTypePair class

```
1 package ch17_6;
2
3 public class TwoTypePair<T1,T2> {
4     private T1 first;
5     private T2 second;
6
7     public TwoTypePair(){
8         first = null;
9         second = null;
10    }
11
12    public TwoTypePair(T1 firstItem, T2 secondItem){
13        first = firstItem;
14        second = secondItem;
15    }
16
17    public void setFirst(T1 first) { this.first = first; }
18
19    public void setSecond(T2 second) { this.second = second; }
20
21    public T1 getFirst() { return first; }
22
23    public T2 getSecond() { return second; }
24
25    public String toString() { return ("first: " + first.toString() + "\n" + "second: " + second.toString()); }
26
27    public boolean equals(Object otherObject){
28        if (otherObject == null) return false;
29        else if(getClass() != otherObject.getClass()) return false;
30        else {
31            TwoTypePair<T1, T2> otherPair = (TwoTypePair<T1, T2>)otherObject;
32            return (first.equals(otherPair.first) && second.equals(otherPair.second));
33        }
34    }
35 }
```

TwoTypePairDemo class

```
1 package ch17_6;
2
3 import java.util.Scanner;
4
5 public class TwoTypePairDemo {
6     public static void main(String[] args) {
7         TwoTypePair<String, Integer> rating = new TwoTypePair<>("The Car Guys", 8);
8         Scanner keyboard = new Scanner(System.in);
9         System.out.println("Our current rating for " + rating.getFirst());
10        System.out.println(" is " + rating.getSecond());
11        System.out.println("How would you rate them?");
12        int score = keyboard.nextInt();
13        rating.setSecond(score);
14        System.out.println("Our new rating for " + rating.getFirst());
15        System.out.println(" is " + rating.getSecond());
16        System.out.println("by 2017250035 이종수");
17    }
18 }
```

실행 결과

```
Run: TwoTypePairDemo x
/Library/Java/JavaVirtualMachines/jd
Our current rating for The Car Guys
is 8
How would you rate them?
10
Our new rating for The Car Guys
is 10
by 2017250035 이종수
Process finished with exit code 0
```

앞선 Song class를 활용하는 예제입니다. 이번에는 두 개의 Generics를 활용합니다. 2 개의 변수 모두 T type으로 선언합니다. 이후 Drive 클래스에서 String, Integer로 하고 "The Car Guys", 8로 설정합니다. 이후 사용자로 부터 입력을 받아 8을 바꿔서 출력합니다(setSecond 사용). 따라서 위와 같은 결과가 출력되는 것을 확인할 수 있습니다.

7.

UnorderedPair class

```
1 package ch17_7;
2
3 import ch17_5.Pair;
4
5 public class UnorderedPair<T> extends Pair<T> {
6     public UnorderedPair(){
7         setFirst(null);
8         setSecond(null);
9     }
10    public UnorderedPair(T firstItem, T secondItem){
11        setFirst(firstItem);
12        setSecond(secondItem);
13    }
14    public boolean equals(Object otherObject){
15        if (otherObject == null) return false;
16        else if (getClass() != otherObject.getClass()) return false;
17        else {
18            UnorderedPair<T> otherPair = (UnorderedPair<T>) otherObject;
19            return (getFirst().equals(otherPair.getFirst()) && getSecond().equals(otherPair.getSecond()))
20                || (getFirst().equals(otherPair.getSecond()) && getSecond().equals(otherPair.getFirst()));
21        }
22    }
23 }
```

UnorderedPairDemo

```
1 package ch17_7;
2
3 public class UnorderedPairDemo {
4     public static void main(String[] args){
5         UnorderedPair<String> p1 = new UnorderedPair<String>("peanuts", "beer");
6         UnorderedPair<String> p2 = new UnorderedPair<String>("beer", "peanuts");
7         if (p1.equals(p2)){
8             System.out.println(p1.getFirst() + " and " + p1.getSecond() + " is the same as");
9             System.out.println(p2.getFirst() + " and " + p2.getSecond());
10            System.out.println("by 2017250035 이종수");
11        }
12    }
13 }
```

실행 결과

```
Run: UnorderedPairDemo x
/Library/Java/JavaVirtualMachines/j
peanuts and beer is the same as
beer and peanuts
by 2017250035 이종수
Process finished with exit code 0
```

5번에서 만든 Pair 클래스를 상속받아 DerivedGeneric 클래스를 만들어 줍니다. 5번과 코드의 차이는 거의 없고, equals() 메소드만 조금 수정합니다. 여기서는 first와 first, second와 second, first와 second, second와 first를 비교하게 합니다. 이후 Drive 클래스에서 "peanuts", "beer"로 p1을 만들고, "beer", "peanuts"로 p2를 만들고 두 개를 비교하여 그 결과를 출력합니다. DeriveGenerics 의 equals()

메소드에 의해 true를 리턴하므로 2개가 같다고 출력됩니다. 따라서 위와 같은 결과가 출력되는 것을 확인할 수 있습니다.