

1. 다음과 같이 Human 인터페이스와 구현 클래스 Worker가 있다. Worker 클래스는 이미 다른 프로젝트에서 사용 중이다. 그런데 Human 인터페이스를 구현하는 Student 클래스에 sleep() 메서드가 필요하게 되었다. 또 Human 타입으로 사용할 echo() 메서드도 필요하다. 따라서 다음과 같은 테스트 프로그램을 실행하고자 한다. Human 인터페이스, Worker 클래스에 수정할 부분이 있으면 수정하고, Student 클래스도 작성하라.

<Human 인터페이스, Worker 클래스>

```
interface Human{
    void eat();
}

class Worker implements Human{
    public void eat() {
        System.out.println("빵을 먹습니다.");
    }
}
```

<실행 결과>

```
빵을 먹습니다.
도시락을 먹습니다.
쿨쿨
야호!
```

<테스트 프로그램>

```
public class HunamTest {
    public static void main(String[] args) {

        Worker w = new Worker();
        w.eat();

        Student s = new Student();
        s.eat();

        s.sleep();

        Human.echo();
    }
}
```

<답안 예시>

```
1 interface Human{
2     void eat();
3     static void echo() {
4         System.out.println("야호!");
5     }
6     default void sleep() {
7         System.out.println("쿨쿨");
8     }
9 }
10
11 class Worker implements Human{
12     public void eat() {
13         System.out.println("빵을 먹습니다.");
14     }
15 }
16
17 class Student implements Human{
18
19     @Override
20     public void eat() {
21         System.out.println("도시락을 먹습니다.");
22     }
23
24 }
```

```
26 public class HunamTest {
27     public static void main(String[] args) {
28
29         Worker w = new Worker();
30         w.eat();
31
32         Student s = new Student();
33         s.eat();
34
35         s.sleep();
36
37         Human.echo();
38     }
39 }
--
```

2. 다음과 같은 인터페이스와 테스트 프로그램 및 실행결과가 있다. 각 항목에서 요구한 사항에 부합하도록 테스트 프로그램을 완성하시오.

<Talkable 인터페이스, 테스트 프로그램>

```
interface Talkable{  
    void talk();  
}
```

```
public class TalkableTest {  
  
    static void conversation(Talkable t) {  
        //코드 추가  
    }  
  
    public static void main(String[] args) {  
        //코드 추가  
        conversation(new Korean());  
        conversation(new American());  
    }  
}
```

<실행 결과>

```
안녕하세요!  
Hello!
```

- ④ Talkable 구현 클래스인 Korean과 American 클래스를 테스트 프로그램의 main() 메서드에 지역 클래스로 정의해 프로그램을 테스트하라.

<코드 예시>

```
1 interface Talkable{  
2     void talk();  
3 }  
4  
5 public class TalkableTest {  
6  
7     static void conversation(Talkable t) {  
8         t.talk();  
9     }  
10  
11     public static void main(String[] args) {  
12         class Korean implements Talkable{  
13             @Override  
14             public void talk() {  
15                 System.out.println("안녕하세요!");  
16             }  
17         }  
18  
19         class American implements Talkable{  
20             @Override  
21             public void talk() {  
22                 System.out.println("Hello!");  
23             }  
24         }  
25         conversation(new Korean());  
26         conversation(new American());  
27     }  
28  
29 }
```

- ② ①에서 작성한 프로그램에서 Korean과 American 클래스를 무명객체로 수정해 main() 메서드를 완성 후 테스트 하라.

<코드 예시>

```
1 interface Talkable{
2     void talk();
3 }
4
5 public class TalkableTest {
6
7     static void conversation(Talkable t) {
8         t.talk();
9     }
10
11     public static void main(String[] args) {
12
13         conversation(new Talkable() {
14             @Override
15             public void talk() {
16                 System.out.println("안녕하세요!");
17             }
18         });
19
20         conversation(new Talkable(){
21             @Override
22             public void talk() {
23                 System.out.println("Hello!");
24             }
25         });
26     }
27 }
28 }
```

- ③ ②에서 작성한 무명 객체를 람다식으로 수정해 테스트 하라.

<코드 예시>

```
1 interface Talkable{
2     void talk();
3 }
4
5 public class TalkableTest {
6
7     static void conversation(Talkable t) {
8         t.talk();
9     }
10
11     public static void main(String[] args) {
12
13         conversation(()->System.out.println("안녕하세요!"));
14
15         conversation(()->System.out.println("Hello!"));
16     }
17 }
18 }
```

3. 키보드로 입력한 정수 값을 학생의 점수로 사용하며 ArrayList에 저장한다. 입력된 정수 값이 음수면 입력의 끝이다. 입력된 점수를 내림차순으로 정렬하고 평균과 함께 출력하라.

<실행 결과>

```
점수를 입력하세요: 68
점수를 입력하세요: 90
점수를 입력하세요: 88
점수를 입력하세요: 75
점수를 입력하세요: 98
점수를 입력하세요: -1
점수(내림차순):
98 90 88 75 68
평균: 83.8
```

<코드 예시>

```
1*import java.util.ArrayList;
4
5 public class ArrayListTest {
6
7     public static void main(String[] args) {
8         ArrayList<Integer> list = new ArrayList<>();
9         Scanner sc = new Scanner(System.in);
10        int input;
11
12        System.out.print("점수를 입력하세요: ");
13
14        while((input = sc.nextInt()) >= 0) {
15            list.add(input);
16            System.out.print("점수를 입력하세요: " );
17        }
18        System.out.println("점수(내림차순): ");
19        list.stream().sorted(Collections.reverseOrder()).forEach((x)->System.out.print(x + " "));
20
21        System.out.println();
22
23        double sum = 0;
24
25        for(int i : list) {
26            sum += i;
27        }
28
29        System.out.println("평균: " + sum / list.size() );
30    }
31
32 }
```

4. `BufferedReader` 스트림을 사용해 키보드에서 행 단위로 읽어 `ArrayList`에 저장한 후 사전순으로 콘솔 뷰에 출력하는 프로그램을 작성하라. 'q'가 입력되면 입력을 종료한다.

<실행 결과>

인터페이스  
추상클래스  
내부클래스  
람다식  
입출력  
q  
내부클래스 람다식 인터페이스 입출력 추상클래스

<코드 예시>

```
1 import java.io.BufferedReader;
2 import java.io.IOException;
3 import java.io.InputStreamReader;
4 import java.util.ArrayList;
5
6 public class BufferedReaderTest {
7
8     public static void main(String[] args) {
9         String str = null;
10        ArrayList<String> list = new ArrayList<>();
11
12        try(InputStreamReader isr = new InputStreamReader(System.in);
13            BufferedReader br = new BufferedReader(isr)){
14
15            while((str = br.readLine()) != null) {
16                if(str.equals("q"))
17                    break;
18                list.add(str);
19            }
20
21        } catch(IOException e) {
22            System.out.println(e.getMessage());
23        }
24
25        list.stream().sorted().forEach((x) -> System.out.print(x + " "));
26    }
27
28 }
```