

## 객체지향프로그래밍 - 12주차 실습 답안지

성명: 조우연

학과: 수학과

학번: 201921195

※ 본 실습활동지에 대한 보고서나 코드를 작성함에 있어 교재나 강의노트를 제외한 다른 자료로부터 일부 또는 전체를 복사하였습니까? 예( ) 아니오( O )

1. 강의노트(ch05 part 5) Listing 5.11.을 수행해본 후 다음 물음에 답하시오.

(가) 3개의 String을 저장하는 ArrayList를 만들어 각 스트링을 출력해보시오.

### - 수정한 코드

```
public class ArrayListTest {
    public static void main(String[] args) {
        ArrayList<Employee>staff=new ArrayList<>();
        staff.add(new Employee("Carl Cracker",75000,1987,12,15));
        staff.add(new Employee("Harry Hacker",50000,1989,10,1));
        staff.add(new Employee("Tony Tester",40000,1990,3,15));

        for(Employee e : staff)
            e.raiseSalary(5);

        ArrayList<String>ToString=new ArrayList<>();
        for(Employee e : staff)
            ToString.add("name=" + e.getName() + ",salary=" + e.getSalary()+",hireDay="+e.getHireDay());

        for(String s : ToString)
            System.out.println(s);

        //for(Employee e : staff)
        // System.out.println("name="+e.getName()+",salary="+e.getSalary()+",hireDay="+e.getHireDay());
    }
}
```

### - 수행결과

```
name=Carl Cracker,salary=78750.0,hireDay=1987-12-15
name=Harry Hacker,salary=52500.0,hireDay=1989-10-01
name=Tony Tester,salary=42000.0,hireDay=1990-03-15
```

(나) (가)에서 만든 ArrayList에서 역순으로 원소를 제거하면서 출력하시오.

- 코드

```
public class ArrayListTest {
    public static void main(String[] args) {
        ArrayList<Employee>staff=new ArrayList<>();
        staff.add(new Employee("Carl Cracker",75000,1987,12,15));
        staff.add(new Employee("Harry Hacker",50000,1989,10,1));
        staff.add(new Employee("Tony Tester",40000,1990,3,15));

        for(Employee e : staff)
            e.raiseSalary(5);

        ArrayList<String>ToString=new ArrayList<>();
        for(Employee e : staff)
            ToString.add("name=" + e.getName() + ",salary=" + e.getSalary()+"hireDay="+e.getHireDay());

        for(int i=ToString.size();i>0;i--){
            String s = ToString.remove(i-1);
            System.out.println(s);
        }

        //for(Employee e : staff)
        // System.out.println("name="+e.getName()+"salary="+e.getSalary()+"hireDay="+e.getHireDay());
    }
}
```

- 실행결과

```
name=Tony Tester,salary=42000.0,hireDay=1990-03-15
name=Harry Hacker,salary=52500.0,hireDay=1989-10-01
name=Carl Cracker,salary=78750.0,hireDay=1987-12-15
```

2. 주어진 String 객체를 저장하는 Stack class를 보고 물음에 답하시오.

(가) Stack class를 이용하여 주어진 코드를 실행해보자. 어떤 문제가 발생하며 왜 발생하는가?

- 설명

오류가 발생한다. 주어진 코드에서는 Stack type의 객체를 생성할 때, 스택의 크기를 2로 설정했다. 스택을 배열 형태로 설정하면 초반에 설정해준 스택의 크기를 넘어가게 되었을 때 ArrayIndexOutOfBoundsException이 발생하게 되어서 Stack을 사용할 수 없게 된다. 즉, (가) 코드를 사용하여 실행하였을 때, 스택의 크기가 2를 벗어났으므로 오류가 발생했다.

(나) (가)의 문제를 해결하기 위해 array 대신에 ArrayList를 이용하여 위 Stack을 다시 구현하고자 한다. 적절하게 고치시오. 수정한 Stack을 이용하여 (가)의 테스트 코드를 실행해보시오.

- 코드	- 실행결과
<pre> public class Stack {     private ArrayList&lt;String&gt;stack;     private int top;     public Stack()     {         stack = new ArrayList&lt;&gt;();     }     public void push(String s) {         stack.add(top, s);         top++;     }     public String pop() {         top--;         return stack.remove(top);     } } </pre>	<pre> &lt;terminated&gt; StackTest [Java Application] C </pre>

(다) 다양한 타입의 데이터를 저장할 수 있도록 (나)에서 작성한 Stack 클래스를 수정하여 generic class Stack을 정의하시오.

- 코드
<pre> public class Stack&lt;T&gt; {     private ArrayList&lt;T&gt;stack;     private int top;     public Stack()     {         stack = new ArrayList&lt;&gt;();     }     public void push(T s) {         stack.add(top, s);         top++;     }     public T pop() {         top--;         return stack.remove(top);     } } </pre>

(라) (다)에서 만든 generic Stack class를 String과 Employee 객체를 이용하여 테스트하십시오.

#### - 코드

```
public class StackTest {

    public static void main(String[] args) {
        Stack<String> ss = new Stack<>();
        Stack<Employee> sc = new Stack<>();

        ss.push("a");
        ss.push("b");
        ss.push("c");
        System.out.println(ss.pop());

        sc.push(new Employee("Jo", 5000, 2020, 11, 18));
        sc.push(new Employee("Woo", 7000, 2020, 04, 17));
        sc.push(new Employee("Yeon", 4000, 2020, 9, 30));
        Employee e = sc.pop();
        System.out.println("name=" + e.getName() + ",salary=" + e.getSalary()+",hireDay="+e.getHireDay());
    }
}
```

#### - 실행결과

```
C
name=Yeon,salary=4000.0,hireDay=2020-09-30
```

3. 주어진 PairTest3를 수행해 본 후 다음 각 질문에 답하십시오.

(가) 주어진 코드를 PairTest3의 main함수의 마지막 부분에 추가하여 컴파일해보시오. 어떤 문제가 발생하는지 왜 발생하는지 설명하십시오.

#### - 실행결과

```
3 public class PairTest3 {
4
5     public static void main(String[] args) {
6         Employee e1 = new Employee("Carl Cracker", 75000, 2007, 12, 15);
7         Employee e2 = new Employee("Gus Greedy", 80000, 2000, 1, 15);
8         Pair<Employee> p = new Pair<>(e1,e2);
9         System.out.println(p.getFirst());
10        System.out.println(p.getSecond());
11
12        Employee e3 = new Employee("Sid Sneaky", 100000, 1989, 1, 1);
13
14        changePartner(p,e3);
15        System.out.println(p.getFirst());
16        System.out.println(p.getSecond());
17
18        Manager m1 = new Manager("Gang Gamchan", 75000, 2007, 12, 15);
19        Manager m2 = new Manager("Yang Manchun", 80000, 2000, 1, 15);
20        Pair<Manager> p2 = new Pair<>(m1,m2);
21        System.out.println(p2.getFirst());
22        System.out.println(p2.getSecond());
23
24        Manager m = new Manager("Lee Sunshin", 100000,1989,1,1);
25
26        changePartner(p2,m);
27        System.out.println(p2.getFirst());
28        System.out.println(p2.getSecond());
29    }
30 }
```

Console Problems Debug Shell

<terminated> PairTest3 [Java Application] C:\Program Files\Java\openjdk-11.0.2\_windows-x64\_bin\jdk-11.0.2\bin\javaw.exe (2020.11.18. 오후 10:33:44 - 오후 10:33:47)

Exception in thread "main" java.lang.Error: Unresolved compilation problem:

The method changePartner(Pair<Employee>, Employee) in the type PairTest3 is not applicable for the arguments (Pair<Manager>, Manager)

at lab\_B\_12\_201921195.PairTest3.main(PairTest3.java:26)

- 설명

실습지에 주어진 코드를 추가하여 실행시키면 위와 같이 컴파일 에러가 발생한다. 왜냐하면 generic class는 상속을 허용하지 않기 때문이다. 즉, Manager는 Employee의 subclass이지만, Pair<Manager>는 Pair<Employee>의 subclass가 아니다. 따라서 changePartner method에 Manager type의 변수를 전달해주면 컴파일 에러를 발생시킨다.

(나) Employee와 Manager 객체의 pair에 정상적으로 작동하기 위해서 changePartner() 함수를 고치시오. 단, changePartner()는 Employee와 모든 subclass의 pair에 작동해야 한다.

- 수정한 코드 (in method of changePartner)

```
public static void changePartner(Pair<? extends Employee> p, Employee e) {  
    p.setSecond(e);  
}
```

- 수정한 코드 (in class of Pair<T>)

```
public class Pair<T>  
{  
    private T first;  
    private T second;  
  
    public Pair() { first = null; second = null; }  
    public Pair(T first, T second) { this.first = first; this.second = second; }  
  
    public T getFirst() { return first; }  
    public T getSecond() { return second; }  
  
    public void setFirst(T newValue) { first = newValue; }  
    //public void setSecond(T newValue) { second = newValue; }  
    public void setSecond(Employee newValue) { second = (T) newValue; }  
}
```

- 실행결과

```
lab_B_12_201921195.Employee@2aae9190  
lab_B_12_201921195.Employee@2f333739  
lab_B_12_201921195.Employee@2aae9190  
lab_B_12_201921195.Employee@77468bd9  
lab_B_12_201921195.Manager@4cc77c2e  
lab_B_12_201921195.Manager@7a7b0070  
lab_B_12_201921195.Manager@4cc77c2e  
lab_B_12_201921195.Manager@39a054a5
```