

## 객체지향프로그래밍 - 12주차 실습 활동지(2020년 11월 18일)

성명: \_\_\_\_\_ 학과: \_\_\_\_\_ 학번: \_\_\_\_\_

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

**제출방식:** 별도의 답안지 파일에 각 문제에 대하여 작성한 답안이나 작성한 코드 및 그 실행 결과를 복사해 넣는다. 답안지 문서명은 "report\_주차\_학번" (예, report\_12\_201811111)로 만든다. 답안지는 pdf형식으로 과제게시판(Bb)에 차주 월요일 23시59분까지 제출한다.

### I. Objectives

1. Generic method를 작성하고 활용할 수 있다.
2. Generic class를 작성하고 활용할 수 있다.
3. Type variable의 upper bound를 이해하고 작성할 수 있다.

### II. Exercises (15점)

1. 강의노트(ch05 part 5) Listing 5.11. (첨부: ArrayListTest.java, Employee.java)을 수행해본 후 다음 물음에 답하시오.

가) 3개의 String을 저장하는 ArrayList를 만들어 각 스트링을 출력해보시오. (코드 및 수행화면) [1점]

나) (가)에서 만든 ArrayList에서 역순으로 원소를 제거하면서 출력하시오. (코드 및 수행화면) [1점]

힌트) remove 메소드를 사용한다.

2. 다음 코드는 String 객체를 저장하는 Stack class이다.

```
public class Stack
{
    private String[] stack;
    private int top;
    public Stack(int n)
    {
        stack = new String[n];
        top = 0;
    }
    public void push(String s)
    {
        stack[top] = s;
        top++;
    }
    public String pop()
```

```

{
    top--;
    return stack[top];
}

```

가) 위 Stack class를 이용하여 아래 코드를 실행해보자. 어떤 문제가 발생하며 왜 발생하는가? (설명) [1점]

```

public class StackTest
{
    public static void main(String[] args)
    {
        Stack s = new Stack(2);
        s.push("a");
        s.push("b");
        s.push("c");
        System.out.println(s.pop());
    }
}

```

나) (가)의 문제를 해결하기 위해 array 대신에 ArrayList를 이용하여 위 Stack을 다시 구현하고자 한다. 적절하게 고치시오. 단, Stack의 생성을 아래와 같은 방식으로 한다. 수정한 Stack을 이용하여 (가)의 테스트 코드를 실행해보시오. (코드 및 실행결과) [3점]

**Stack s = new Stack(); // Stack의 크기를 제공하지 않음**

다) 아래 선언에서와 같이 다양한 타입의 데이터를 저장할 수 있도록 (나)에서 작성한 Stack 클래스를 수정하여 generic class Stack을 정의하시오. (코드) [3점]

```

Stack<String> ss = new Stack<>();
Stack<Employee> sc = new Stack<>();

```

라) (다)에서 만든 generic Stack class를 String과 Employee 객체를 이용하여 테스트하시오. (가)번의 테스트 형식으로 간단하게 테스트하면 됨. (코드 및 실행결과) [2점]

3. 아래 PairTest3는 두 명의 Employee 객체의 짝(pair)으로부터 한 명의 Employee 객체를 변경하는 프로그램이다. changePartner() 함수는 Employee pair에서 두 번째 employee를 주어진 parameter로 바꾸는 함수이다. 아래 프로그램을 수행해 본 후 다음 각 질문에 답하시오. (단, Pair.java, Employee.java, Manager.java를 활용).

```

public class PairTest3
{

```

```

public static void main(String[] args)
{
    Employee e1 = new Employee("Carl Cracker", 75000, 2007, 12, 15);
    Employee e2 = new Employee("Gus Greedy", 80000, 2000, 1, 15);
    Pair<Employee> p = new Pair<>(e1,e2);
    System.out.println(p.getFirst());
    System.out.println(p.getSecond());

    Employee e3 = new Employee("Sid Sneaky", 100000, 1989, 1,1);

    changePartner(p, e3);
    System.out.println(p.getFirst());
    System.out.println(p.getSecond());
}

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

```

가) 아래 코드를 PairTest3의 main함수의 마지막 부분에 추가하여 컴파일해보시오. 어떤 문제가 발생하는지 왜 발생하는지 설명하시오. **(설명) [2점]**

```

Manager m1 = new Manager("Gang Gamchan", 75000, 2007, 12, 15);
Manager m2 = new Manager("Yang Manchun", 80000, 2000, 1, 15);
Pair<Manager> p2 = new Pair<>(m1,m2);
System.out.println(p2.getFirst());
System.out.println(p2.getSecond());

Manager m = new Manager("Lee Sunshin", 100000, 1989, 1,1);

changePartner(p2, m);
System.out.println(p2.getFirst());
System.out.println(p2.getSecond());

```

나) Employee와 Manager 객체의 pair에 정상적으로 작동하기 위해서 changePartener()함수를 고치시오. 단, changePartener()는 Employee와 모든 subclass의 pair에 작동해야 한다. **(코드 및 실행결과) [2점]**