객체지향프로그래밍및실습 – 10주차 실습활동지(11월6일)

성명: _____ 학과: _____ 학번: _____ 실습반: _____

제출방식: 보고서를 pdf 형식으로 제출한다(report10.pdf).

제출기한: 11월 7일(금) 23시59분까지 (지연 제출 0점 처리함, 미완성이라도 반드시 제출하길 바람)

I. Objectives

1. Lambda expression의 개념을 이해하고 표현할 수 있다.
2. Lambda expression과 interface와의 관계를 이해한다.
3. Lambda expression을 용도를 이해하고 활용할 수 있다.

II. Exercises (15점)

1. 다음 각 문항을 위한 lambda exrepssion과 lambda expression의 타입으로 적합한 functional interface를 정의하시오.
[lambda expression을 포함한 functional interface 코드를 보고서에 캡쳐하지 않고 직접 작성][3점]

1-가) 하나의 double 형 값을 받아 제곱을 계산하여 반환

1-나) LocalDate 객체와 정수값 하나를 받아 LocalDate의 연도가 정수값보다 작으면 true를 아니면 false를 반환

1-다) 두개의 스트링을 받아 그 스트링의 길이가 5보다 크면 첫 5문자를 반환하고 아니면 그 스트링을 그대로 반환

2. Employee 클래스(첨부파일 Employee.java)를 이용하여 물음에 답하시오.

2-가) 아래의 EmployeeSortTest는 Employee 객체를 급여(salary) 순(오름차순)으로 정렬하는 프로그램이다. 컴파일해 본 후 어떤 문제가 있는지 설명하시오.
[문제점 설명][2점]

```
import java.util.*;

public class EmployeeSortTest {
    public static void main(String[] args) {
        List<Employee> elist = new ArrayList<>();
        elist.add(new Employee("Lee", 10000));
        elist.add(new Employee("Kim", 20000));
        elist.add(new Employee("Park", 8000));
        elist.add(new Employee("Han", 15000));
```

```
        Collections.sort(elist);
        System.out.println(elist);
    }
}
```

2-나) (가)번의 문제점을 Comparator 인터페이스를 구현하는 lambda expression을 이용하여 해결하고자 한다. 어디를 어떻게 수정하면 되는가? 단, Employee 클래스는 수정할 수 없다.
[EmployeeSortTest.java에서 수정한 코드 첨부][2점]

2-다) 아래의 HowManyEmployeeTest는 Predicate<T> 인터페이스를 이용하여 Employee 배열에서 특정 조건을 만족하는 객체의 개수를 파악하는 코드이다. a) ~ b)의 조건을 만족하도록 빈 곳에 적합한 lambda expression을 각각 채우시오.
[HowManyEmployeeTest에서 수정한 코드 첨부][4점]

```
public class HowManyEmployeeTest {
   public static void main(String[] args) {
      Employee[] emps = new Employee[3];

      emps[0] = new Employee("Kim", 10000);
      emps[1] = new Employee("Lee", 20000);
      emps[2] = new Employee("Kim", 30000);

      int n = HowManyEmployeeTest.howMany(emps,                    ); // a)
      int m = HowManyEmployeeTest.howMany(emps,                    ); // b)
      System.out.println("Number of High - salaried Employees = "+n);
      System.out.println("Number of Employees named Kim = "+m);
    }
   public static int howMany(Employee[] emps, Predicate < Employee > t) {
       int n = 0;
       for (Employee e: emps) {
          if (t.test(e)) {
             n++;
          }
       }
       return n;
    }
}
```

2-(다)-a) salary가 20000 이상인 Employee 객체의 수

2-(다)-b) Kim이라는 이름을 가진 Employee 객체의 수

3. Lambda expression을 이용하여 다음과 같은 형식을 가진 수식(prefix expression)을 하나 읽어 들여 계산하는 프로그램을 작성하고자 한다.

+ 10 20
- 30 5

* 3 5

아래의 ExpressionTest는 인터페이스 Expression과 메소드 f를 이용하여 세 가지 연산(+, -, *)을 계산하는 코드이다. 물음에 답하시오.

```
interface Expression {
    int eval(int a, int b);
}
public class ExpressionTest {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        char op = in.next().charAt(0);
        int x = in.nextInt();
        int y = in.nextInt();
        Expression exp = null;

        switch (op) {
            (A)
        }
        int result = (B)  ;
        System.out.println("result = " + result);
    }
    public static int f(Expression e, int a, int b) {
        return e.eval(a, b);
    }
}
```

가) (A)부분에 op에 따라 exp 변수에 적절한 lambda exrepssion을 지정하는 코드를 작성하시오.
[작성한 (A)부분 코드 첨부][2점]

나) (B)부분에 (A)의 exp을 계산하는 수식을 작성하시오.
[작성한 (B)부분 코드 첨부][2점]