

객체지향프로그래밍 - 10주차 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. String의 길이를 기준으로 다음 두가지 방식으로 정렬하는 코드를 작성하시오.

- LengthComparator class를 정의하여 정렬(짧은 것->긴 것 으로 정렬)

```
public class LengthComparator implements Comparator<String> {
    public int compare(String first, String second)
    {
        return first.length()-second.length();
    }
}
```

- lambda expression을 이용해 (1)의 역순으로 정렬(WordSortTest에서 생성)

```
import java.util.Arrays;

public class WordSortTest {

    public static void main(String[] args) {
        String[] names= {"Lee","Chang","Hong"};

        //sorting with LengthComparator
        Arrays.sort(names, new LengthComparator());
        for(String name : names)
            System.out.println(name);

        System.out.println();

        //sorting with lambda expression
        Arrays.sort(names, (first,second)->second.length()-first.length());
        for(String name : names)
            System.out.println(name);
    }
}
```

- 실행결과

<pre>Lee Hong Chang</pre>	위 : 1번 방법 실행결과 아래 : 2번 방법 실행결과
<pre>Chang Hong Lee</pre>	

2. 구분구적법을 이용하여 적분을 계산하는 프로그램을 수행하고 물음에 답하시오.

(가) main함수에서 다음 각 함수를 lambda expression으로 정의하여 적분 값을 구해보시오. 단, 괄호 속은 각각 a, b, n값임. (소수점 이하 5번째 자리까지만 출력한다.)

- 문제에 주어진 적분값을 구하기 위한 main함수 코드

- import java.math.*: 사용하였음.

```
public static void main(String[] args)
{
    F f;
    double r;
    //f = x->x*x;
    f = x->x;
    r = integral(f,0,10,100);
    System.out.printf("f(x)=x, 0, 10, 100: %.5f\n",r);

    f = x->Math.sin(x);
    r = integral(f,0,Math.PI,100);
    System.out.printf("f(x)=sin(x), 0, PI, 100: %.5f\n",r);
}
```

- 수행결과

```
f(x)=x, 0, 10, 100: 50.00000
f(x)=sin(x), 0, PI, 100: 2.00008
```

(나) 0~PI 구간의 sin(x)적분값은 2이지만, (가)에서 구한 값은 2의 근사치가 나온다. 2가 되는 n의 최솟값을 구하여라. 단, 소수점 이하 5번째 자리까지만 사용.

- n을 100부터 499까지 변경시키면서 n에 따른 적분값과 n값을 출력하는 코드

```
public static void main(String[] args)
{
    F f;
    double r=0.0;

    f = x->Math.sin(x);
    int n;
    for(n=100; n<500; n++) {
        r = integral(f,0,Math.PI,n);
        System.out.printf("f(x)=sin(x), 0, PI, %d: %.5f\n",n, r);
    }
}
```

- 수행결과

```
f(x)=sin(x), 0, PI, 403: 2.00001  
f(x)=sin(x), 0, PI, 404: 2.00001  
f(x)=sin(x), 0, PI, 405: 2.00001  
f(x)=sin(x), 0, PI, 406: 2.00000  
f(x)=sin(x), 0, PI, 407: 2.00000  
f(x)=sin(x), 0, PI, 408: 2.00000  
f(x)=sin(x), 0, PI, 409: 2.00000  
f(x)=sin(x), 0, PI, 410: 2.00000
```

- 403,404,405... 이 부분이 n 값.

- 2.00000이 되는 n 의 최솟값은 406이다.

- $n=100\sim 499$ 까지의 결과값이 모두 출력되었지만, 적분값이 2.00000이 나오는 부분외의 앞 뒤 출력결과는 생략하였다.

3. 강의노트 ch06-part5의 Listing 6.7을 수행해본 후 다음 물음에 답하시오.

(가) TalkingClock의 start 메소드 내에서 TimePrinter 객체를 생성하여 Timer 객체에 등록하는 작업을 InnerClassTest class의 main 함수에서 하도록 변경하여 동일한 결과가 나오도록 수정하시오.

- 수정된 start method(in TalkingClock class)

```
public void start(TimePrinter listener)  
{  
    Timer t = new Timer(interval, listener);  
    t.start();  
}
```

- 수정된 main 함수

```
public class InnerClassTest {  
    public static void main(String[] args) {  
        TalkingClock clock = new TalkingClock(1000,true);  
  
        TalkingClock.TimePrinter listener = clock.new TimePrinter();  
        clock.start(listener);  
  
        JOptionPane.showMessageDialog(null,"Quit program?");  
        System.exit(0);  
    }  
}
```

- 수행결과

```
At the tone, the time is Thu Nov 05 22:29:03 KST 2020  
At the tone, the time is Thu Nov 05 22:29:04 KST 2020  
At the tone, the time is Thu Nov 05 22:29:05 KST 2020  
At the tone, the time is Thu Nov 05 22:29:06 KST 2020  
At the tone, the time is Thu Nov 05 22:29:07 KST 2020  
At the tone, the time is Thu Nov 05 22:29:08 KST 2020  
At the tone, the time is Thu Nov 05 22:29:09 KST 2020  
At the tone, the time is Thu Nov 05 22:29:10 KST 2020  
At the tone, the time is Thu Nov 05 22:29:11 KST 2020  
At the tone, the time is Thu Nov 05 22:29:12 KST 2020
```

4. 강의노트 ch06-part5의 Listing 6.8의 anonymous inner class를 수행해본 후 다음 두 가지 방식으로 수정하여 실행하시오.

- local inner class방식으로 수정한 TalkingClock class를 TalkingClock2로, lambda expression 방식으로 수정한 TalkingClock class를 TalkingClock3로 만들어 수행하였습니다.

(가) local inner class

(AnonymousInnerClassTest 의 main함수 코드는 강의노트와 동일하게 진행)

- 코드

```
class TalkingClock2 {  
    public void start(int interval, final boolean beep)  
    {  
        class TimePrinter implements ActionListener{  
            public void actionPerformed(ActionEvent event){  
                System.out.println("At the tone, the time is "+new Date());  
                if(beep)Toolkit.getDefaultToolkit().beep();  
            }  
        }  
        ActionListener listener = new TimePrinter();  
        Timer t = new Timer(interval, listener);  
        t.start();  
    }  
}
```

- 수행결과

```
At the tone, the time is Wed Nov 04 22:01:34 KST 2020  
At the tone, the time is Wed Nov 04 22:01:35 KST 2020  
At the tone, the time is Wed Nov 04 22:01:36 KST 2020  
At the tone, the time is Wed Nov 04 22:01:37 KST 2020  
At the tone, the time is Wed Nov 04 22:01:38 KST 2020  
At the tone, the time is Wed Nov 04 22:01:39 KST 2020  
At the tone, the time is Wed Nov 04 22:01:40 KST 2020  
At the tone, the time is Wed Nov 04 22:01:41 KST 2020  
At the tone, the time is Wed Nov 04 22:01:42 KST 2020  
At the tone, the time is Wed Nov 04 22:01:43 KST 2020  
At the tone, the time is Wed Nov 04 22:01:44 KST 2020  
At the tone, the time is Wed Nov 04 22:01:45 KST 2020  
At the tone, the time is Wed Nov 04 22:01:46 KST 2020  
At the tone, the time is Wed Nov 04 22:01:47 KST 2020  
At the tone, the time is Wed Nov 04 22:01:48 KST 2020  
At the tone, the time is Wed Nov 04 22:01:49 KST 2020
```



-> 우측의 OK누르면 종료됨

(나) lambda expression

(AnonymousInnerClassTest 의 main함수 코드는 강의노트와 동일하게 진행)

- 코드

-> 아래에 코드있음!


```

class TalkingClock2 {
    public void start(int interval, final boolean beep)
    {
        class TimePrinter implements ActionListener{
            public void actionPerformed(ActionEvent event){
                System.out.println("At the tone, the time is "+new Date());
                if(beep)Toolkit.getDefaultToolkit().beep();
            }
        }
        ActionListener listener = new TimePrinter();
        Timer t = new Timer(interval, listener);
        t.start();
    }
}

public class TalkingClock3 {
    public void start(int interval, boolean beep)
    {
        ActionListener listener = new ActionListener() {
            public void actionPerformed(ActionEvent event) {
                System.out.println("At the tone, the time is "+ new Date());
                if(beep) Toolkit.getDefaultToolkit().beep();
            }
        };
        Timer t = new Timer(interval, event->
        {
            System.out.println("At the tone, the time is "+ new Date());
            if(beep) Toolkit.getDefaultToolkit().beep();
        });
        t.start();
    }
}

```

- 수행결과

```

At the tone, the time is Wed Nov 04 22:18:37 KST 2020
At the tone, the time is Wed Nov 04 22:18:38 KST 2020
At the tone, the time is Wed Nov 04 22:18:39 KST 2020
At the tone, the time is Wed Nov 04 22:18:40 KST 2020
At the tone, the time is Wed Nov 04 22:18:41 KST 2020
At the tone, the time is Wed Nov 04 22:18:42 KST 2020
At the tone, the time is Wed Nov 04 22:18:43 KST 2020
At the tone, the time is Wed Nov 04 22:18:44 KST 2020
At the tone, the time is Wed Nov 04 22:18:45 KST 2020
At the tone, the time is Wed Nov 04 22:18:46 KST 2020
At the tone, the time is Wed Nov 04 22:18:47 KST 2020
At the tone, the time is Wed Nov 04 22:18:48 KST 2020

```