

## 객체지향 프로그래밍 기말고사 (2021-12-13)

1.  $1+(-2)+3+(-4)+\dots$  과 같은 식으로 계속 더해 나갔을 때, 몇까지 더해야 총합이 100이상이 되는지

Ex1 클래스를 통해 프로그램을 작성하시오. (10점)

\* 출력의 예

num=199

sum=100

2. StringBuffer 메소드를 사용하여 출력과 같은 결과가 나오도록 Ex2 클래스를 완성하시오.

StringBuffer의 메소드를 사용하여 결과를 보기와 같이 출력한다. (15점)

```
public class Ex2 {
    public static void main(String[] args) {
        StringBuffer sb;
        ...
    }
}
```

\* 출력 예

capacity : 10

StringBuffer : Strawberry Grape Pear Apple

capacity : 27

StringBuffer : Strawberry Grape

capacity : 16

StringBuffer : Strawberry Pear Grape

capacity : 34

StringBuffer : Apple Pear Grape

capacity : 34

3. Ex3 클래스의 main() 메소드와 실행결과를 참고하여, Shape 인터페이스를 구현한 클래스 Circle, Oval, Rect를 작성하여 프로그램을 완성하시오. (15점)

```
interface Shape {
    final double PI = 3.14;
    void draw();
    double getArea();
}

public class Ex3 {
    static public void main(String [] args) {
        Shape [] list = { new Circle(20),
            new Oval(20, 30), new Rect(20, 30)};
        for (Shape a : list)
            a.draw();
        for (Shape a : list)
            System.out.println("면적은 " + a.getArea());
    }
}
```

반지름이 20인 원입니다.  
30x40에 내접하는 타원입니다.  
20x30크기의 사각형입니다.  
면적은 1256.0  
면적은 1884.0000000000002  
면적은 600.0

4. Queue 자료구조를 제네릭 클래스로 선언하고, String과 Integer 큐를 각각 사용하는 예를 Ex4 클래스에서 보여주고 있다. 주어진 Ex4 클래스와 결과내용을 참고하여 GQueue 제네릭 클래스를 작성하시오.

(컬렉션 프레임워크를 사용할 수 없다.) (20점)

<pre>public class Ex4 {     public static void main(String[] args) {         int size=10;          GQueue&lt;String&gt; stringQueue = new GQueue&lt;&gt;(size);         stringQueue.enqueue("seoul");         stringQueue.enqueue("busan");         stringQueue.enqueue("LA");          for(int i=0; i&lt;size; i++) {             String str = stringQueue.dequeue();             if(str != null) System.out.println(str);         }          GQueue&lt;Integer&gt; intQueue = new GQueue&lt;&gt;(size);         intQueue.enqueue(1);         intQueue.enqueue(2);         intQueue.enqueue(3);          for(int i=0; i&lt; size; i++) {             Integer is = intQueue.dequeue();             if(is != null) System.out.println(is);         }     } }</pre>	<pre>* 출력 예 seoul busan LA 1 2 3</pre>
---	--

5. Ex5 클래스는 Student 클래스를 통해 학생들의 시험정보를 관리하고 정렬한다.

출력 예에서 보면 정렬 1 은 성적 총점순이다. 이와 같은 정렬이 될 수 있도록 Student 클래스를 작성하시오. 그리고 정렬 2 와 같이 이름순으로 정렬하기 위해 빈칸에 Comparator 인터페이스의 compare 메소드를 정의하는 람다식을 완성하시오. (20 점)

<pre> public class Ex5 {     public static void main(String[] args) {         ArrayList&lt;Student&gt; list = new ArrayList&lt;&gt;();         list.add(new Student("이순신", 100, 100, 100));         list.add(new Student("홍길동", 90, 70, 80));         list.add(new Student("김철수", 80, 80, 90));         list.add(new Student("이영희", 70, 90, 70));         list.add(new Student("안자바", 60, 100, 80));         Collections.sort(list);          System.out.println("정렬 1");         for(Student s : list)             System.out.println(s);          System.out.println("정렬 2");         Collections.sort(list, _____);         for(Student s : list)             System.out.println(s);     } } </pre>	<p>* 출력 예</p> <p>- 정렬 1</p> <p>이영희 : 230 (70,90,70)  홍길동 : 240 (90,70,80)  안자바 : 240 (60,100,80)  김철수 : 250 (80,80,90)  이순신 : 300 (100,100,100)</p> <p>- 정렬 2</p> <p>김철수 : 250 (80,80,90)  안자바 : 240 (60,100,80)  이순신 : 300 (100,100,100)  이영희 : 230 (70,90,70)  홍길동 : 240 (90,70,80)</p>
--	---

6. Ex6 클래스는 5명의 이름을 ArrayList에 저장하고 이 중에서 "D"로 시작하는 이름은 제외한다. 그리고 모든 이름을 대문자로 변경한다. 이에 대해 아래와 같이 2개의 메소드에 람다식을 이용하여 출력과 같이 결과가 되도록 완성하시오. (20점)

<p>1) 대문자로 "D"로 시작하는 이름은 리스트에서 제외한다.  default boolean removeIf(Predicate&lt;? super E&gt; filter)  - 람다식 : "D"로 시작하는 이름인지를 검증한다.</p> <p>2) 리스트의 이름을 모두 대문자로 변경한다.  static void replaceAlpha(UnaryOperator&lt;String&gt; u, List&lt;String&gt; list) { ... }  - 람다식 : 이름을 대문자로 변경한다.</p>	<p>[David, Thomas, Bill, Eric, Ford]  [Thomas, Bill, Eric, Ford]  [THOMAS, BILL, ERIC, FORD]</p>
---	--