기술인증테스트 Java개발 실기형 문제지

[2017년 3차]

사번	성명		
	1. 공정한 평가를 위해 동료를 도와주는 행위, 보여주는 행위를 금지하고 있습니다.		
유의 사항	2. 부정행위 적발 시, 응시한 평가는 0점 처리됩니다.		
3. 본 시험지는 응시장 외부로 유출할 수 없으며, 시험 종료 후 감독관에게 제출해야 합니다.			



! 실기테스트 구현 시 주의사항

- ✓ 별도 제공된 실기가이드에 따라 실습 환경을 구성합니다.
- ✓ 실습 환경을 구성하고, 제공된 파일을 eclipse에서 작동하는 것도 시험의 일부입니다. 실기가이드를 따라 하시면 됩니다. 환경 구성에 대해 감독관은 질문을 받지 않습니다.
- ✓ 본 실기테스트는 JDK 7 버전을 기준으로 작성합니다.
- ✓ 문항에 대한 질문은 받지 않습니다. 단, 오탈자 및 문항오류에 대해서는 문의하실 수 있습니다.
- ✓ 제공된 클래스와 메소드의 이름 및 형태를 변경할 수 없습니다.
 - → 채점 툴(tool)을 통한 자동 채점이 이루어집니다. 제공된 클래스와 메소드의 원형을 변경하시면 채점이 되지 않습니다.
- ✓ 구현 내용은 반드시 주어진 메소드 내에서 작성해야 합니다.
- ✓ 구현 시, 필요하다면 클래스와 메소드, 변수를 추가로 정의할 수 있습니다.

1. Cube Sum

개요

NXN 정사각 큐브가 존재한다. 주어진 큐브를 시계방향으로 90도 회전하여 두 개의 합을 구하는 프로그램을 작성한다.

문항설명

3X3으로 구성된 큐브가 주어진다. 주어진 큐브를 시계방향으로 90도 회전한다.

큐브(변수명:int[][] cube)

5	7	8
4	2	1
2	5	6



2	4	5
5	2	7
6	1	8

주어진 큐브와 시계방향으로 90도 회전한 큐브의 합을 구한다.

큐브의 합(타입: int[][] 2차원배열)

7	11	13
9	4	8
8	6	14

제공 파일

- ✓ 제공된 프로젝트(LCP_JavaTest)를 import하여 알맞은 클래스에 내용 구현 ※ 프로젝트 import 방법은 "실기 가이드" 참고
- √ Java Class
 - com.lgcns.test.cubesum.CubeSumRun
 - : CubeSum 프로그램의 시작 / 출력 기능을 담당하는 클래스 (제공)
 - com.lgcns.test.cubesum.CubeSum
 - : int [][] sum(int [][] cube) 큐브의 합을 계산하는 메소드 (구현)

배점

아래 제시된 기능의 구현 완성도에 따라 점수를 부여함 (단, 컴파일 에러 발생 시 점수가 부여되지 않으며, 제시된 요구사항과 다를 경우 감점 발생)

1. 기능의 정확성 (10점)

실행 예시

큐브의 합: 7 11 13 9 4 8 8 6 14

2. 사원정보 출력

개요

주어진 정보를 이용하여 사원정보를 출력하는 프로그램을 작성한다.

문항설명

부서정보와 사원정보가 존재하며 부서코드를 통해서 상호 정보를 참고할 수 있다.

- 부서정보는 부서코드., 부서명으로 구성된다.
- 사원정보는 사원번호, 사원명, 부서코드로 구성된다.
- 사원의 부서명을 알고 싶을 경우에는 부서코드를 참고해야 한다.

예) 사원 김길동(사원번호:37123, 부서코드:10)은 영업팀(부서코드:10)이다.

부서정보(변수명 : department)

부서코드	부서명
10	영업팀
20	구매팀
30	개발팀
40	운영팀

사원정보(변수명: employee)

사원번호	사원명	부서코드
59545	최선명	10
23561	박선주	20
37123	김길동	10
33777	이기주	30

요구사항

- 1. 부서명, 사원번호, 사원명으로 구성된 사원정보를 출력하시오.
- 2. 출력순서는 부서코드의 오름차순으로 정렬한다.
- 3. 동일부서의 사원정보는 사원번호의 오름차순으로 정렬한다.

부서명	사원번호	사원명
영업팀	37123	김길동
영업팀	59545	최선명
구매팀	23561	박선주
개발팀	33777	이기주

제공 파일

- ✓ 제공된 프로젝트(LCP_JavaTest)를 import하여 알맞은 클래스에 내용 구현※ 프로젝트 import 방법은 "실기 가이드" 참고
- ✓ Java Class
 - com.lgcns.test.employee.EmployeeRun
 - : Employee 프로그램의 시작 / 출력 기능을 담당하는 클래스 (제공)
 - com.lgcns.test.employee.Employee
 - :String [][] print(String [][] department, String[][] employee) 메소드(구현)

배점

아래 제시된 기능의 구현 완성도에 따라 점수를 부여함

(단, 컴파일 에러 발생 시 점수가 부여되지 않으며, 제시된 요구사항과 다를 경우 감점 발생)

1. 기능의 정확성(16점)

실행 예시

사원정보 출력

[부서명, 사원번호, 사원명]

[영업팀, 37123, 김길동]

[영업팀, 59545, 최선명]

[구매팀, 23561, 박선주]

[개발팀, 33777, 이기주]

3. Dense Ranking

개요

학생정보에서 점수가 높은 순서대로 랭킹을 구하는 프로그램을 작성한다.

요구사항

- 1. 점수가 높은 순서대로 랭킹을 구한다. (단, 동점일 경우 그 다음 순위는 누락되지 않는다)
- 2. 동일한 랭킹 내에서는 학번의 오름차순으로 출력한다.

[INPUT]

학번	점수
10210	80
10211	77
10212	55
10213	66
10214	77
10215	99
10216	30
10217	10

[OUTPUT]

학번	점수	랭킹	
10215	99	1	
10210	80	2	
10211	77	3	
10214	77	3	동점자
10213	66	4 -	→ 4등 포함
10212	55	5	
10216	30	6	
10217	10	7	

제공 파일

- ✓ 제공된 프로젝트(LCP_JavaTest)를 import하여 알맞은 클래스에 내용 구현 ※ 프로젝트 import 방법은 "실기 가이드" 참고
- ✓ Java Class
 - com.lgcns.test.denserank.DenseRankRun
 - : 점수가 높은 순으로 정렬하여 랭킹을 구하는 프로그램의 시작 / 출력 기능을 담당하는 클래스 (제공)
 - com.lgcns.test.denserank.DenseRank
 - :public static int[][] rank(int[][] input)
 - . 점수가 높은 순으로 정렬하여 랭킹을 구하는 메소드 (구현)
 - . parameter는 {학번, 점수} array이며, return은 {학번, 점수, 랭킹} array여야 한다.

배점

아래 제시된 기능의 구현 완성도에 따라 점수를 부여함 (단, 컴파일 에러 발생 시 점수가 부여되지 않으며, 제시된 요구사항과 다를 경우 감점 발생)

1. 기능의 정확성(16점)

실행 예시

4. Source Analyzer

개요

프로젝트에서 성능테스트 하던 도중 소스의 몇 가지 패턴에서 문제가 발견되었다. 이러한 문제들이 미치는 영향도를 분석하기 위한 프로그램을 작성하려고 한다.

문항 설명

영향도 분석 규칙은 아래와 같다.

[영향도 분석 대상 소스]

- 1. 영향도 분석 대상 소스의 위치는 디렉토리(변수명 dirName)의 하위 파일들이다. 디렉토리 위치 : "C:\java_test\workspace\LCP_JavaTest\resource\sourceanalyzer"
- 2. 해당 디렉토리 하위에 디렉토리는 없는 것으로 간주한다.
- 3. Java소스만을 대상으로 영향도를 분석한다.(.java 이외의 파일은 무시한다)

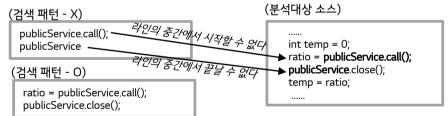
[검색 패턴 규칙]

- 1. 검색 패턴은 String 값으로 1개가 주어진다.(변수명: pattern, 값은 변경 가능하다)
- 2. 검색 패턴은 1줄 이상의 라인으로 구성될 수 있다.
- 3. "공백"은 검색 패턴과 분석대상 소스에서 모두 무시하고 영향도를 분석한다.(탭은 고려하지 않는다.) 예) "ratio = ratio + 2"와 "ratio=ratio+2"는 동일한 것으로 간주
- 4. "빈 줄"과 "줄 바꿈"는 검색 패턴과 분석대상 소스에서 모두 무시하고 영향도를 분석한다.
 - 예) 아래의 검색 패턴은 분석대상 소스와 동일한 것으로 간주하여



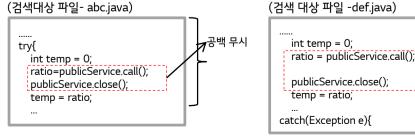
5. 검색 패턴은 소스 라인 중간에서 시작하거나, 중간에서 끝나지 않는다.

예) 아래와 같은 검색 패턴은 고려하지 않는다.(단어가 아닌 줄 단위로만 검색이 가능하다)



[검색 예] (검색 패턴)

ratio = publicService.call(); publicService.close();



- 검색 결과: 문제가 되는 java 소스 파일은 2개이며 검색된 패턴 수는 2개임.



빈 줄 무시

요구사항

다음과 같은 분석 결과를 출력하는 영향도 분석 프로그램을 작성하시오.

- 전체 소스 개수 : 주어진 디렉토리 하위에 존재하는 java 파일 개수
- 문제 소스 개수 : 검색 패턴이 발견된 java 파일 개수
- 소스 전체 Line 수 : 전체 java 파일의 소스 line 합계 (빈 줄과 공백으로만 구성된 빈 줄은 제외)
- 검색된 패턴 수 (줄 수 아님)

제공 파일

- ✓ 제공된 프로젝트(LCP_JavaTest)를 import하여 알맞은 클래스에 내용 구현 ※ 프로젝트 import 방법은 "실기 가이드" 참고
- ✓ Java Class
 - com.lgcns.test. Sourceanalyzer. SourceAnalyzerRun
 - : 프로그램의 시작 / 출력 기능을 담당하는 클래스 (제공)
 - com.lgcns.test.sourceanalyzer.SourceAnalyzer
 - :List(String) readFile(File file)
 - . 파일의 내용을 line 단위로 읽어서 String List로 리턴하는 메소드 (제공)
 - :void analyze(String dirName, String pattern) 영향도를 분석하는 메소드(구현)
 - . 분석 결과는 아래 4개의 메소드를 통해 반환해야 한다.
 - : int getJavaFileCount() 전체 소스 개수를 반환하는 메소드 (구현)
 - :int getProblemFileCount() 문제 소스 개수를 반환하는 메소드 (구현)
 - : int getTotalLineCount() 소스 전체 Line 수를 반환하는 메소드 (구현)
 - : int getPatternCount() 검색된 패턴수를 반환하는 메소드 (구현)
- ✓ 테스트를 위한 샘플 디렉토리
 - data/sourceanalyzer/dir1
 - data/sourceanalyzer/dir2

배점

아래 제시된 기능의 구현 완성도에 따라 점수를 부여함

(단, 컴파일 에러 발생 시 점수가 부여되지 않으며, 제시된 요구사항과 다를 경우 감점 발생)

- 1. 전체 소스 개수의 정확성 (3점)
- 2. 문제가(검색 패턴이) 발견된 소스 개수의 정확성 (6점)
- 3. 소스 전체 Line 수의 정확성 (3점)
- 4. 검색된 패턴 수의 정확성 (10점)

실행 예시

대상 디렉토리 : data/sourceanalyzer/dir1

검색 패턴: public void remove() {removeDescending();}

[분석 결과]

전체 소스 개수 : 5 문제 소스 개수 : 1

소스 전체 Line 수 : 5221

검색된 패턴수: 2



수고하셨습니다.

