

기술인증테스트

Java개발-A

실기형 문제지

[2017년 17차]

사번	성명
유의 사항	<ol style="list-style-type: none">1. 공정한 평가를 위해 동료들 도와주는 행위, 보여주는 행위를 금지하고 있습니다.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, 오타자 및 문항오류에 대해서는 문의하실 수 있습니다.
- ✓ 답안 제출 시스템(평가시스템, JCTEST)은 제한 시간이 지나면 자동으로 종료됩니다.
반드시 시간 안에 제출할 수 있도록 시간 관리를 잘 해 주셔야 합니다.
- ✓ 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. Word Count

개요

단어(Word)들로 구성된 문자열에서 검색 단어의 빈도 수를 출력하는 프로그램을 작성하시오.

요구사항

- 문자열은 다음과 같이 구성되진다.
 - 문자열은 Word로 구성되어 있으며, Word사이에는 앞과 뒤에 공백이 한 개 존재한다.
 - 문자열의 첫 Word의 앞과, 마지막 Word의 뒤에는 공백이 없다.
 - Word는 영문자, 숫자, 특수문자로 구성된다.

ex) I am a boy. → 다음 4개의 Word로 구성되어 있다("I", "am", "a", "boy.")

ex) I LOVE LG LG CNS^^ → 다음 4개의 Word로 구성되어 있다("I", "LOVE", "LG", "LG", "CNS^^")
- 문자열에서 검색 Word의 빈도 수를 구하여 반환한다.

ex) 문자열 : I LOVE <u>LG</u> <u>LG</u> CNS^^	ex) 문자열 : I LOVE LG LG CNS^^
검색단어 : LG	검색단어 : CNS
빈도 수 : 2회	빈도 수 : 0회

제공 파일

- ✓ 제공된 프로젝트(LCP_JavaTest)를 import하여 알맞은 클래스에 내용 구현
 - ※ 프로젝트 import 방법은 “실기 가이드” 참고
- ✓ Java Class
 - com.lgcns.test.wordcount.WordCountRun
 - : 검색 단어의 빈도 수를 구하는 프로그램의 시작과 출력기능을 담당하는 클래스 (제공)
 - com.lgcns.test.wordcount.WordCount
 - : public int countWord(String message, String searchWord)
입력받은 message에서 searchWord의 빈도수를 구하여 결과를 리턴한다.(구현)
 - . 매개변수 : String message – 입력받은 문자열
 - . 매개변수 : String searchWord – 검색 단어
 - . 리턴 : 검색단어의 빈도 수(int)

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

입력된 문자열 : I LOVE LG LG CNS^^
 검색 단어 : LG
 빈도 수 : 2

입력된 문자열 : I LOVE LG I LOVE CNS
 검색 단어 : I
 빈도 수 : 2

2. 직급별 평균연봉

개요

주어진 정보를 이용하여 직급별 평균연봉 정보를 출력하는 프로그램을 작성한다.

문항설명

연봉정보와 직원정보가 존재하며 부서코드를 통해서 상호 정보를 참고할 수 있다.

- 연봉정보는 직원번호, 연봉(단위 : 만원) 으로 구성된다.
- 직원정보는 직원번호, 직원명, 직급(사원, 대리, 과장, 차장, 부장)으로 구성된다.
- 직원의 연봉을 알고 싶을 경우에는 연봉정보를 참고해야 한다.

예) 과장 모모(직원번호: 23561)의 연봉은 5600만원이다.

연봉정보(변수명 : salary)

직원번호	연봉(만원)
23561	5600
37123	4100
33777	4400
13451	6300
23579	5900
87594	2400
59545	3500
49376	3600

직원정보(변수명 : employee)

직원번호	직원명	직급
59545	나연	대리
37123	정연	대리
23561	모모	과장
33777	사나	과장
23579	지효	차장
87594	미나	사원
13451	다현	부장
49376	채영	대리

요구사항

1. 직급, 평균연봉으로 구성된 직급별 평균 연봉정보를 출력하시오.
- 출력순서는 평균연봉의 오름차순이다.
 - 평균연봉은 만원 단위에서 반올림하여 처리한다.

ex) 대리 평균연봉

$$(3500 + 3600 + 4100) / 3$$
$$= 3733.3333 \rightarrow (\text{반올림}) 3733 \text{ 만원}$$

직급	평균연봉(만원)
사원	2400
대리	3733
과장	5000
차장	5900
부장	6300

제공 파일

- ✓ 제공된 프로젝트(LCP_JavaTest)를 import하여 알맞은 클래스에 내용 구현
 - ※ 프로젝트 import 방법은 “실기 가이드” 참고
- ✓ Java Class
 - com.lgcns.test.salary.SalaryRun
 - : 직급별 평균연봉을 계산하는 프로그램의 시작 / 출력 기능을 담당하는 클래스 (제공)
 - com.lgcns.test.salary.Salary
 - : String [][] print(String [][] salary, String[][] employee)
 - . 직급별 평균연봉을 출력하는 메소드(구현)
 - . 매개변수 : salary : 연봉정보
 - . 매개변수 : employee : 직원정보
 - . 리턴 : String [][] : 직급별 평균연봉 정보

배점	<div>아래 제시된 기능의 구현 완성도에 따라 점수를 부여함</div> <div>(단, 컴파일 에러 발생 시 점수가 부여되지 않으며, 제시된 요구사항과 다를 경우 감점 발생)</div> <div>1. 평균연봉 계산의 정확성(10점)</div> <div>2. 평균연봉의 오름차순 정렬(6점)</div>
실행 예시	<div>직급별 평균연봉 정보</div> <div>[직급, 평균연봉(만원)]</div> <div>[사원, 2400]</div> <div>[대리, 3733]</div> <div>[과장, 5000]</div> <div>[차장, 5900]</div> <div>[부장, 6300]</div>

3. MAC주소 검증 및 변환하기

개요	<p>MAC주소란 네트워크 인터페이스에 할당된 고유 식별자이다.</p> <p>MAC주소를 입력 받아 올바른 형식인지 검증하고 하나의 형식으로 변환하는 프로그램을 작성한다.</p>
문항 설명	<p>MAC주소는 총 48bit(6byte)로 구성되어 있으며, 다음 3가지 형식으로 표시될 수 있다.</p> <p>[Type 1] 하이픈(-)으로 구분된 2자리 16진수로 구성된 6개의 그룹</p> <p>ex) 01-23-45-67-89-ab</p> <p>[Type 2] 콜론(:)으로 구분된 2자리 16진수로 구성된 6개의 그룹</p> <p>ex) 01:23:45:67:89:ab</p> <p>[Type 3] 점(.)으로 구분된 4자리 16진수로 구성된 3개의 그룹</p> <p>ex) 0123.4567.89ab</p> <p>주어진 MAC주소가 위 3가지 형식 중 하나에 부합하는지를 검증하여 부합한다면 [Type 3] 형식으로 변환하여 반환하고 부합하지 않으면 null을 반환한다.</p>
요구 사항	<ol style="list-style-type: none">16진수는 0~9의 숫자와 A~F의 문자 중 하나로 표시된다. 검증 시에는 대소문자는 구분 하지 않고, 변환 시에는 소문자로 변환한다. ex) 01-23-45-67-89-AB -> 0123.4567.89ab (Type 1에 부합, Type 3으로 변환)주어진 MAC주소가 Type1, Type2에 부합한다면 Type 3로 변환하고, Type 3 형식에 부합한다면 소문자로 변환 후, 반환하면 된다. ex) 01:23:45:67:89:ab -> 0123.4567.89ab (Type 2에 부합, Type 3으로 변환) ex) 0123.4567.89ab -> 0123.4567.89ab (Type 3에 부합)주어진 MAC주소가 어떤 형식에도 부합하지 않으면 null을 반환한다. ex) 01-23-45:67-89-ab -> null (-와 :를 혼용해서 사용하였음) ex) 01-23-45-67-89-gh -> null (16진수가 아님)
제공 파일	<ul style="list-style-type: none">✓ 제공된 프로젝트(LCP_JavaTest)를 import하여 알맞은 클래스에 내용 구현 ※ 프로젝트 import 방법은 "실기 가이드" 참고✓ Java Class<ul style="list-style-type: none">- com.lgcns.test.macaddress.MacAddressRun : MacAddress 프로그램의 시작 / 출력 기능을 담당하는 클래스 (제공)- com.lgcns.test.macaddress.MacAddress : public String validateMacAddress(String inputMac) → 입력받은 MAC주소를 검증하고 Type 3 형식으로 변환하여 리턴한다 (구현) . 매개변수 : String inputMac : 입력 MAC주소 . 리턴 : String : 변환된 MAC주소, 올바르지 않은 형식은 null을 리턴

배점

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

- 1. MAC주소 검증의 정확성 (8점)
- 2. MAC주소 변환의 정확성 (6점)
- 3. 대소문자 처리의 정확성 (2점)

실행 예시

inputMac : 01:23:45:67:89:ab
올바른 MAC 주소입니다.
0123.4567.89ab

inputMac : 0123.AbcD.Ef99
올바른 MAC 주소입니다.
0123.abcd.ef99

inputMac : 78-90-AB-CD-EF-GH
올바르지 않은 MAC 주소입니다.

4. 단어 찾기

개요

알파벳 대문자로 채워진 사각형 모양의 격자에서 입력 받은 단어를 찾는 검색 프로그램을 작성한다.

요구사항

입력 받은 단어가 격자 내에 존재하면 시작 좌표를 출력하는 프로그램을 작성하시오.

1. 사각형 격자의 모양은 항상 10 X 10 정사각형이다.

2. 격자의 내용은 알파벳 대문자로만 구성된다.

3. 가로좌표는 0~9까지 숫자로 표현하며, 세로 좌표는 a~j 까지 영문소문자로 표현한다.

4. 입력 받은 단어는 알파벳 대문자로만 구성된다. (두자리 이상의 단어만 입력된다.)

5. 단어는 가로, 세로 또는 대각선으로 찾는다. 단어의 방향은 아래와 같다.

- 가로 : 좌 → 우

- 세로 : 상 → 하

- 대각선 : (좌, 상) → (우, 하)

6. 입력 받은 단어가 격자 내에 존재할 경우 시작좌표를 리턴한다.

- 리턴 값은 시작좌표(String) 배열이다.

7. 입력 받은 단어가 격자 내에 존재하지 않으면 null을 리턴한다.

좌표	0	1	2	3	4	5	6	7	8	9		
a	B	H	E	Z	R	B	U	O	C	Z		
b	L	S	L	Z	K	E	Q	B	S	A		
c	A	Y	P	M	D	E	D	A	Q	Y		
d	C	W	R	P	G	D	I	N	D	I		
e	K	F	U	I	R	T	Y	K	N	N		
f	B	G	P	R	E	D	G	E	W	W		
g	L	G	I	Y	E	R	O	V	U	O		
h	U	N	Y	L	N	P	U	X	W	R		
i	K	V	O	E	S	I	T	B	C	B		
j	I	W	N	Z	W	H	I	T	E	Y		

찾을 단어

시작좌표

BLACKa0

REDa4, f3

GREENd4

BROWNull

WHITEj4

BLUenull

ORANGEnull

- 제공 파일
- ✓ 제공된 프로젝트(LCP_JavaTest)를 import하여 알맞은 클래스에 내용 구현

※ 프로젝트 import 방법은 “실기 가이드” 참고

✓ Java Class

- com.lgcns.test.wordsearch.WordSearchRun

: Word Search 프로그램의 시작 / 출력 기능을 담당하는 클래스 (제공)

- com.lgcns.test.wordsearch.WordSearch

: public String [] findWordCount(String [][] matrix, String word)

. 해당 단어가 격자 내에 존재 하는 경우 시작좌표를 반환하는 메소드 (구현)

. 매개변수 : matrix : 알파벳 대문자가 포함된 격자

. 매개변수 : word : 격자 내에서 찾을 단어

. 리턴 : String [] : 입력 받은 단어가 격자 내에 존재할 경우 시작좌표

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 - 가로 방향의 문자열 검색 (6점)
- 2. 기능의 정확성 - 세로 방향의 문자열 검색 (6점)
- 3. 기능의 정확성 - 대각선 방향의 문자열 검색 (10점)

실행 예시

검색 단어 목록 : [BLACK, RED, GREEN, BROWN, WHITE, BLUE, ORANGE]

검색 결과
단어 : BLACK, 시작좌표 : [a0]
단어 : RED, 시작좌표 : [a4,f3]
단어 : GREEN, 시작좌표 : [d4]
단어 : BROWN, 시작좌표 : 없음
단어 : WHITE, 시작좌표 : [j4]
단어 : BLUE, 시작좌표 : 없음
단어 : ORANGE, 시작좌표 : 없음

수고하셨습니다.

