**Report04**



|  |  |
| --- | --- |
| **과 목 :** | **소프트웨어프로젝트** |
| **제출 일자 :** | **2019.06.02** |
| **담당 교수 :** | **이남규** |
| **학 과 :** | **사회복지학부** |
| **학 번 :** | **20170299** |
| **이 름 :** | **김유빈** |

1. 사용자로부터 입력과 출력 파일명을 입력 받고, 입력 파일에 들어 있는 텍스트의 문자수, 단어수, 문장수를 count하여 출력하는 프로그램을 쓰시오. 출력결과는 출력 파일에 저장되도록 한다.

<해결할 문제>

1) 입력과 출력 파일명 입력 받고 파일 객체 생성

2) 파일 입력 스트림 생성

3) 입력 받은 문자 배열(buf)로 문자수, 단어수, 문장수 계산

4) 파일 출력 스트림 생성

5) 계산한 결과를 출력 파일에 저장(쓰기)

**1) 입력과 출력 파일명 입력 받고 파일 객체 생성**

Scanner scanner = **new** Scanner(System.***in***);

String f1 = scanner.nextLine(); //입력 파일

String f2 = scanner.nextLine(); //출력 파일

입력을 받고

File src = **new** File("C:\\Users\\mk\\eclipse-workspace\\Report4\\" + f1);

File dest = **new** File("C:\\Users\\mk\\eclipse-workspace\\Report4\\" + f2);

읽을 파일(src)와 출력 파일(dest) 객체 생성

**2) 파일 입력 스트림 생성**

예외 처리 하고(try – catch)

FileReader fr = **new** FileReader(src);

**3) 입력 받은 문자 배열(buf)로 문자수, 단어수, 문장수 계산**

①문자수(공백을 제외)

**char** buf[] = **new** **char**[1024];

버퍼를 생성하여 읽은 텍스트 파일의 모든 문자열을 buf에 저장

c = fr.read(buf); // c는 실제 읽은 문자수(공백까지 포함)

charcount = c;

read 함수는 읽은 char형의 개수를 반환한다

따라서 charcount(문자수)에 c를 저장

단, 문자수는 공백을 제외해야 하므로

buf 문자 배열을 탐색하면서 공백 수만큼 빼줘야한다!

**if**(buf[i] == ' ') //공백이 아닌 모든 문자

charcount--;

②단어수

단어는 단어 다음에 있는 공백 . ? ! 의 개수를 새는 것과 마찬가지이다.

**for**(**int** i=0;i<buf.length;i++) {

**if**(buf[i] == ' ' || buf[i] == '.' || buf[i] == '!' || buf[i] == '?') {

wordcount++;

단, 문장에서 . ! ? 다음에 공백이 나오므로 이를 제외시켜야함★

**if**(buf[i] == ' ' && (buf[i-1] == '.' || buf[i-1] == '!' || buf[i-1] == '?'))

wordcount--;

③문장수

**if**(buf[i] == '.' || buf[i] == '!' || buf[i] == '?')

stringcount++;

문장 : 세가지 문자(. ? ! )로 끝나면 하나의 문장이다.

**4) 파일 출력 스트림 생성**

FileWriter fw = **new** FileWriter(dest);

**5) 계산한 결과를 출력 파일에 저장(쓰기)**

int형인 문자수, 단어수, 문장수를 string형으로 바꿔서 파일에 출력한다

String str1 = Integer.*toString*(charcount);

String str2 = Integer.*toString*(wordcount);

String str3 = Integer.*toString*(stringcount);

이렇게 문자열로 바뀐 변수들을 파일 출력 스트림을 통해 파일에 저장

fw.write(str1,0,str1.length()); //문자수 저장

fw.write("\r\n",0,2); //엔터키 저장

fw.write(str2,0,str2.length()); //단어수 저장

fw.write("\r\n",0,2);

fw.write(str3,0,str3.length()); //문장수 저장

<출력결과>

-입력-

스크린샷이(가) 표시된 사진

자동 생성된 설명

-출력 파일 결과화면-

스크린샷이(가) 표시된 사진

자동 생성된 설명

2. 다음과 같은 GUI 모양을 가진 스윙 프레임을 작성하라 10개의 ‘\*’ 문자는 10개의 JLabel을 이용하여 랜덤한 위치에 출력한다.

<핵심 문제>

1) 스윙 프레임 작성 – 컨테이너 생성

2) 스윙 프레임 작성 – 패널 3개 생성

3) 최종 컨테이너에 패널 담기

**1) 스윙 프레임 작성 – 컨테이너 생성**

**public** **class** Report04\_2 **extends** JFrame

setTitle, setVisible, setSize 설정

Container c = getContentPane();

c.setLayout(**new** BorderLayout());

패널 3개를 최종 컨테이너에 담기 위해

c를 생성한다

패널을 south, north, center로 배치하기 위해 레이아웃을 BorderLayout

**2) 스윙 프레임 작성 – 패널 3개 생성**

JPanel panel1 = new JPanel();

panel1.setBackground(Color.*gray*);

panel1.add(new JButton("열기"));

패널1: 배경색(회색), 버튼 생성한다

패널2: 10개의 \*문자가 랜덤한 위치에서 보이는 패널이다

랜덤한 숫자 출력을 위해 Random 객체 생성

랜덤하게 컴포넌트를 위치시키기 위해 배치관리자가 없는 컨테이너 생성

JPanel panel2 = **new** JPanel();

panel2.setLayout(**null**);

Random random = **new** Random(); //랜덤 숫자 만들기

**for**(**int** i = 0;i<10;i++) {

JLabel label = **new** JLabel("\*");

label.setSize(10,10); //\*\*이거 안하면 문자가 안보임!!!

label.setLocation(random.nextInt(250),random.nextInt(250));

panel2.add(label);

}

label.setLocation(random.nextInt(250),random.nextInt(250));

->문자\*를 랜덤한 위치에 배정하는 부분!

label.setSize(10,10);

->배치관리자가 있으면 배치관리자가 알아서 창에 맞춰서 크기와 위치를 조절하지만

배치관리자가 없으면 크기가 초기화(0)인 상태이므로

크기를 정해주지 않으면 컴포넌트가 보이지 않는다!!

JPanel panel3 = **new** JPanel();

panel3.setBackground(Color.***yellow***);

panel3.add(**new** JTextField(20));

패널3: 텍스트필드 크기를 20으로 주었다!

**3) 최종 컨테이너에 패널 담기**

c.add(panel1,BorderLayout.***NORTH***);

c.add(panel2,BorderLayout.***CENTER***);

c.add(panel3,BorderLayout.***SOUTH***);

최종 컨테이너에 패널을 담아야지 패널이 화면에 출력된다.

패널만 만들면 패널이 창에 담아지지 않는다!!

<출력화면>

스크린샷이(가) 표시된 사진

자동 생성된 설명

3. 이벤트처리 프로그램: “C”를 출력하는 JLabel 을 하나 만들고, 초기 위치를 (100, 100)으로 하고, 마우스로 “C”를 클릭할 때마다 컨텐츠팬 내에 랜덤한 위치로 움직이도록 하라

C

클릭

다음 두 개의 라벨

**1) 문자C와 말풍선 라벨 생성**

JLabel label = **new** JLabel("C");

label.setSize(10,10);

label.setLocation(100,100);

초기 위치를 (100, 100)

ImageIcon image = **new** ImageIcon("D:\\중앙대학교 3-1\\소프트웨어프로젝트\\과제4\\image.png");

JLabel label2 = **new** JLabel(image);

label2.setSize(100,100);

label2.setLocation(label.getX() - 30 ,label.getY() - 90);

c.add(label2);

라벨 이미지 만드는 방법

경로에 대한 이미지를 라벨에 넣기

말풍선 라벨의 위치는 문자C라벨의 위치에서 변형시킴

**2) 마우스 이벤트 리스너 생성**

label.addMouseListener(**new** MouseAdapter(){ //마우스 어뎁터 익명 클래스 생성

Random random = **new** Random(); //랜덤한 숫자

**public** **void** mouseClicked(MouseEvent e) { //클릭 이벤트

JLabel b = (JLabel)e.getSource();

b.setSize(10,10);

b.setLocation(random.nextInt(250),random.nextInt(250));

label2.setLocation(label.getX() - 30 ,label.getY() - 90);

}

});

마우스를 클릭할 때 마다 이벤트 소스(문자C라벨)의 위치가 변화되고

그에 따라 말풍선 라벨(label2)의 위치도 같이 변화시키게끔 한다!!!

<출력화면>

-마우스 클릭 전

스크린샷이(가) 표시된 사진

자동 생성된 설명

-마우스 클릭 후

스크린샷이(가) 표시된 사진

자동 생성된 설명