

[과제에 사용되는 것들]

1

- 배열의 선언과 생성
 - ▣ 메소드에서 배열의 리턴
- continue, break 문
- main 메소드 인자 전달 : String [] args
- 자바의 예외처리 : try-catch문

- **String 관련 주요 메소드**
- **ASCII 텍스트 파일에서 개행문자 처리**
- **파일 입출력 방법**

유용한 String 메소드

2

□ JAVA API 참조 사이트

<https://docs.oracle.com/en/java/javase/14/docs/api/java.base/module-summary.html>

→ 패키지 `Java.lang` → 클래스 `String` → 메소드

□ String 관련 주요 메소드

- ▣ boolean `equals`(Object anObject)
- ▣ int `compareTo`(String anotherString)
- ▣ String `substring`(int beginIndex)
- ▣ int `length`()
- ▣ 문자열의 연결 : + 연산자 활용

유용한 String 메소드

3

- ▣ boolean **equals**(Object anObject)
String str1 = "ABC";
String str2 = "BCD";
if (str1.equals(str2)) {
 System.out.println("Same string!");
}
- ▣ int **compareTo**(String anotherString)
if (str1.compareTo(str2) > 0) {
 System.out.println("str1 is after str2");
}
- ▣ String **substring**(int beginIndex)
System.out.println(str1.substring(1));
- ▣ int **length**()
System.out.println(str1.length());
- ▣ 문자열의 연결 : **+** 연산자 활용
String str3 = str1 + str2;
System.out.println(str3);

ASCII 텍스트 파일에서 개행문자 처리

4

- ASCII 텍스트 파일에서는 영문자나 숫자, 그리고 comma, period 등의 보이는 특수문자 외에 보이지 않는 특수문자들이 있다.
 - 개행문자 즉, new line을 표현하는 특수문자로서, CR, LF가 있다.
 - ▣ CR (carrige return) : 프로그램에서는 '~~W~~r'로 표현
 - ▣ LF (line feed) : 프로그램에서는 '~~W~~n'로 표현
- 입력 파일에서 1개의 문자씩 읽어 들일 경우, 라인의 마지막에는 CR과 LF의 2개 문자가 존재하므로 프로그래밍할 때 유의해야 한다.

FileReader을 이용한 파일 읽기

5

□ 파일 전체를 읽어 화면에 출력하는 코드 샘플

test.txt 파일(이클립스 프로젝트 폴더 밑에 존재)을 열고 파일과 입력 바이트 스트림 객체 fin 연결

```
FileReader fin = new FileReader ("test.txt");
```

```
int c;
```

```
while((c = fin.read()) != -1) {
```

파일 끝까지 바이트씩 c에 읽어 들임.
파일의 끝을 만나면 read()는 -1 리턴

```
    System.out.print((char)c);
```

바이트 c를 문자로 변환하여 화면에 출력

```
}
```

```
fin.close();
```

스트림을 닫음. 파일도 닫힘.
스트림과 파일의 연결을 끊음.
더 이상 스트림으로부터 읽을 수 없음

예제 8-1 : FileReader로 텍스트 파일 읽기

6

FileReader를 이용하여 c:\windows\system.ini 파일을 읽어 화면에 출력하는 프로그램을 작성하라. system.ini는 텍스트 파일이다.

```
import java.io.*;
```

```
public class FileReaderEx {  
    public static void main(String[] args) {  
        FileReader fin = null;  
        try {  
            fin = new FileReader("c:\\windows\\system.ini");  
            int c;  
            while ((c = fin.read()) != -1) { // 한 문자씩 파일 끝까지 읽기  
                System.out.print((char)c);  
            }  
            fin.close();  
        }  
        catch (IOException e) {  
            System.out.println("입출력 오류");  
        }  
    }  
}
```

파일의 끝을 만나면 read()는 -1 리턴

```
; for 16-bit app support  
[386Enh]  
woafont=dosapp.fon  
EGA80WOA.FON=EGA80WOA.FON  
EGA40WOA.FON=EGA40WOA.FON  
CGA80WOA.FON=CGA80WOA.FON  
CGA40WOA.FON=CGA40WOA.FON
```

```
[drivers]  
wave=mmdrv.dll  
timer=timer.dr
```

```
[mci]
```

FileWriter 사용 예

7

□ test.txt로의 문자 출력 스트림 생성

```
FileWriter fout = new FileWriter("test.txt");
```

□ 파일 쓰기

▣ 문자 단위 쓰기

```
FileWriter fout = new FileWriter("test.txt");  
fout.write('A'); // 문자 'A' 출력  
fout.close();
```

▣ 블록 단위 쓰기

```
char [] buf = new char [1024];  
  
// buf[] 배열의 처음부터 배열 크기(1024개 문자)만큼 쓰기  
fout.write(buf, 0, buf.length);
```

```
String str = "ABCDEFGH";  
  
// str 문자열 처음부터 쓰기  
fout.write(str, 0, str.length());
```

예제 8-4 : 키보드 입력을 파일로 저장하기

8

Scanner를 이용하여 키보드에서 입력받은 데이터를 test.txt 파일에 저장하는 프로그램을 작성하라.

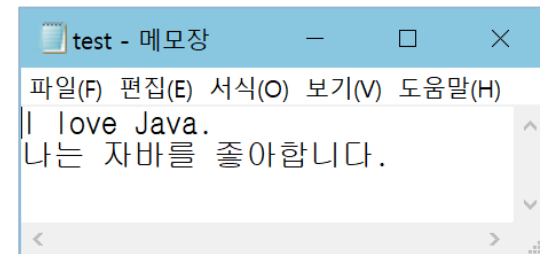
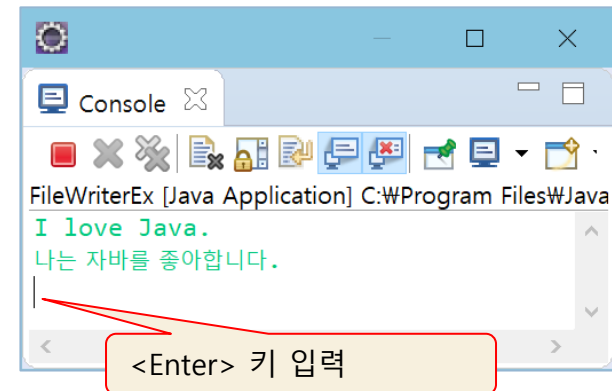
```
import java.io.*;
import java.util.*;

public class FileWriterEx {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        FileWriter fout = null;
        int c;
        try {
            fout = new FileWriter("test.txt");

            while(true) {
                String line = scanner.nextLine();
                if(line.length() == 0)
                    break;
                fout.write(line, 0, line.length());
                fout.write("\r\n", 0, 2);
            }

            fout.close();
        } catch (IOException e) {
            System.out.println("입출력 오류");
        }
        scanner.close();
    }
}
```

한 줄 띄기 위해
WrWn을 파일에 저장



실행 결과 test.txt 파일 생성

과제 설계(1) – 처리 구조

9

입출력 파일 open

주요 변수 선언 및 초기화

(반복) 1문자 읽기 -> 문단내 라인수 or 0 / 문단카운트 10미만

- (반복) 문단 단위 처리

- ▣ (반복) 라인단위 처리

- (반복) 문자 단위 처리

- ▣ 해당 문단의 결과를 찾고, REPORT 출력

파일 close

과제 설계(2) – Tips

10

- 파일의 입출력은 try-catch() 필요
- 문자 단위 처리 details
 - 최대 20개까지 반복하여 1개 문자를 읽고,
 - 읽어낸 문자가
 - Ngram 체크에 포함할 문자인가? → 발견된 ngram 등록하고, 해당 카운트 증가
 - 공백, 쉼표, 마침표, 개행문자('\n')인가? → 반복문 계속 진행
 - 개행문자('\n')인가? → 라인 내 반복 끝
- 주요 배열 (참고용)
 - `String [] uniArr = new String[200];`
 - `String [] biArr = new String[200];`
 - `String [] triArr = new String[200];`

 - `int [] uniCount = new int[200];`
 - `int [] biCount = new int[200];`
 - `int [] triCount = new int[200];`

과제 동작 원리 예시

11

3

a a. a,a.

bc bc

abcd abcd abcd

Paragraph #1

unigram : **a**

bigram : **bc**

trigram : **abc**

tempUni

"a"

tempBi

"a"

tempTri

"a"

uniArr

"a"

"b"

"c"

"d"

""

uniCount

7

5

5

3

0

biArr

"bc"

"ab"

"cd"

""

biCount

5

3

3

0

triArr

"abc"

"bcd"

""

triCount

3

3

0

""

0

""

0

""

0