



- 입출력 스트림은 입출력 장치와 프로그램 사이의 일련의 데이터 흐름을 의미한다.
- 문자 스트림은 유니코드로 된 문자 단위의 데이터가 흐르는 스트림이다. 문자가 아닌 정보는 문자 스트림의 데이터가 될 수 없다.
- 문자 스트림을 다루는 자바의 모든 클래스는 추상 클래스인 `Reader`와 `Writer` 클래스의 서브 클래스이다.
- 문자 스트림은 문자 정보를 가진 텍스트 파일만 처리할 수 있다. 바이너리 파일의 입출력에는 사용할 수 없다.
- 문자 입력 스트림을 다루는 자바 클래스에는 `Reader`, `InputStreamReader`, `FileReader` 등이 있다.
- 문자 출력 스트림을 다루는 자바 클래스에는 `Writer`, `OutputStreamWriter`, `FileWriter` 등이 있다.
- 바이트 스트림은 8비트 바이트 단위의 데이터가 흐르는 스트림이다.
- 바이트 스트림을 다루는 자바의 모든 클래스는 추상 클래스인 `InputStream`과 `OutputStream`의 서브 클래스이다.
- 바이트 스트림은 텍스트 파일의 입출력에도 사용 가능하지만 이미지나 동영상과 같은 바이너리 데이터를 입출력하는 데 보다 적합하다.
- 바이트 입력 스트림을 다루는 자바 클래스에는 `InputStream`, `FileInputStream`, `DataInputStream` 등이 있다.
- 바이트 출력 스트림을 다루는 자바 클래스에는 `OutputStream`, `FileOutputStream`, `DataOutputStream` 등이 있다.
- 버퍼 입출력(Buffered I/O)은 입출력 스트림과 프로그램 사이에 버퍼를 두어 입출력 장치의 속도와 프로그램의 처리 속도의 완충 작용을 수행하므로 효율적으로 입출력을 처리한다.
- 바이트 버퍼 스트림을 다루는 자바 클래스는 `BufferedInputStream`과 `BufferedOutputStream`이 있다.
- 문자 버퍼 스트림을 다루는 자바 클래스는 `BufferedReader`와 `BufferedWriter`가 있다.
- `File` 클래스는 파일과 디렉터리 경로명의 추상적 표현이다.
- `File` 클래스는 파일 삭제, 파일 이름 변경, 디렉터리 생성 등과 같은 파일 관리 작업을 수행한다.
- `File` 클래스는 파일 입출력을 위한 기능은 제공하지 않는다.



행맨(HangMan) 게임 만들기

Open Challenge

목·적

파일 읽기, 벡터 다루기

행맨 게임을 만들면서 파일 입출력을 연습해보자. 행맨은 많이 알려져 있는 전통 있는 게임이다. 25143개의 영단어를 가진 `words.txt` 파일을 읽고, 사용자 모르게 영어 단어 하나를 선택한다. 그리고 이 단어에서 몇 개의 글자를 숨긴 다음 화면에 출력하여 사용자가 단어를 맞추게 하는 게임이다. 숨긴 글자의 수가 많으면 그만큼 난이도가 높다. 여기서는 2개의 글자만 숨기도록 한다. 한 단어에 대해 5번 틀리면 프로그램을 종료한다. `words.txt` 파일은 독자들에게 배포된 자료에 들어 있으며, 한 줄에 하나의 영어 단어가 적혀 있다. **난이도 7**

지금부터 행맨 게임을 시작합니다.

o-colog-

>>y

o-cology

>>n

oncology

Next(y/n)?y

b-ckpl-ne

>>a

backplane

Next(y/n)?y

n-dula-

>>k

n-dula-

>>o

nodula-

>>s

nodula-

>>p

nodula-

>>w

nodula-

>>b

5번 실패 하였습니다.

nodular

Next(y/n)?

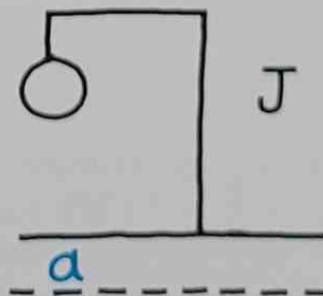
힌트

- words.txt 파일의 모든 단어를 읽어 다음의 스트링 벡터에 저장한다.

```
Vector<String> wordVector = new Vector<String>();
```

- words.txt 파일에는 한 라인에 하나의 영어 단어가 들어 있으며, 다음과 같이 Scanner를 이용하여 한 라인씩 읽을 수 있다.

```
Scanner scanner = new Scanner(new FileReader("words.txt"));  
while(scanner.hasNext()) { // 파일의 끝까지 반복하여 읽는다.  
    String word = scanner.nextLine(); // 한 라인 단위로 읽는다. word에 '\n'은 없음  
    wordVector.add(word); // 단어를 벡터에 저장한다.  
}
```



연습문제

EXERCISE



이론문제

• 출수 문제는 정답이 공개됩니다.

1. 스트림에 대한 설명으로 틀린 것은?
 - ① 입출력이 동시에 되는 스트림은 없다.
 - ② 스트림은 다른 스트림과 연결될 수 없다.
 - ③ 스트림은 먼저 들어온 데이터를 먼저 내보내는 방식이다
 - ④ 스트림은 바이트를 다루는 스트림과 문자만 다루는 스트림으로 나뉘어진다.

2. 다음 응용에서 어떤 입출력 스트림 클래스를 사용하여야 하는가?

(1) 음악 파일 연주	(2) PPT 파일 복사
(3) 영어와 한글이 기록된 사전 읽기	(4) 선수 이름과 점수를 문자열로 저장

3. 이미지 파일을 읽으려고 한다. 가장 적합한 클래스는?

① InputStreamReader	② File
③ FileReader	④ FileInputStream

4. 다음은 파일을 읽기 위해 스트림을 생성하는 코드이다. 이 코드가 실행될 때 발생할 수 있는 예외가 어떤 것이 있는지 설명하고, try-catch 블록으로 감싸는 코드를 작성하라.


```
FileInputStream fin = new FileInputStream("song.mp3");
```

5. 파일의 크기를 알려고 할 때 필요한 클래스는?

① File	② FileSize
③ System	④ FileInputStream

6. File 클래스의 직접적인 기능이 아닌 것은?

① 파일 읽고 쓰기	② 파일 이름 변경
③ 새 디렉터리 만들기	④ 파일 삭제

7. 다음 코드에 대해 물음에 답하라.

```
File file = new File("c:\\windows\\system.ini");
```

- (1) file.isFile()의 리턴 값은?
- (2) file.getParent()의 리턴 값은?
- (3) file.getPath()의 리턴 값은?
- (4) file.getName()의 리턴 값은?
- (5) 아래 빈칸을 채워 문제의 코드와 동일한 코드를 구성하라.

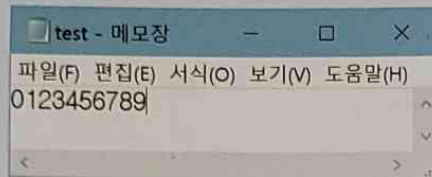
```
File file = new File(_____, "system.ini");
```

8. 문자 집합에 대한 설명으로 틀린 것은?

- ① 문자들의 인코딩 정보로 구성된 문자 인코딩 세트를 뜻한다.
- ② 텍스트 파일 읽기 시 문자 집합을 알려주지 않으면 제대로 읽지 못하는 경우가 발생한다.
- ③ 바이트 스트림 입출력 시 문자 집합을 고려하여 입출력해야 한다.
- ④ 윈도우에서 작성된 텍스트 파일은 기본적으로 "MS949" 문자 집합으로 인코딩된다.

9. c:\temp\test.txt 파일을 읽어 화면에 출력하는 코드이다. 빈칸을 완성하라.

```
FileInputStream fin = null;
try {
    fin = new FileInputStream(_____);
    int c;
    while(true) {
        c = fin.read(); // 파일에서 한 바이트 읽기
        if (_____) break; // 파일 끝까지 읽었음
        System.out.print((char)c);
    }
    _____; // 파일 입력 스트림 닫기
} catch (_____) {
    System.out.println("파일을 찾을 수 없습니다.");
} catch (IOException e) {
    System.out.println("입출력 오류가 발생했습니다.");
}
```



0123456789

10. 다음은 50 바이트 크기의 버퍼를 사용하여 C:\temp\sample.txt를 읽어 화면에 출력하는 소스 코드의 일부이다. 빈칸을 채워라.

```
BufferedOutputStream bout = _____;
File f = new File(_____);
FileReader fin = new FileReader(f);
int c;
while (_____) {
    _____;
}
fin.close();
bout.close();
```

실습문제

• 출수 문제는 정답이 공개됩니다.

FileWriter로 텍스트 파일 저장

1. Scanner로 입력받은 이름과 전화번호를 한 줄에 한 사람씩 c:\temp\phone.txt 파일에 저장하라. "그만"을 입력하면 프로그램을 종료한다. 난이도 4

```
전화번호 입력 프로그램입니다.
이름 전화번호 >> 황기태 010-5555-7777
이름 전화번호 >> 이재문 011-3333-4444
이름 전화번호 >> 김남윤 065-2222-1111
이름 전화번호 >> 그만
c:\temp\phone.txt에 저장하였습니다.
```

2. 앞서 저장한 c:\temp\phone.txt 파일을 읽어 화면에 출력하라. 난이도 4

🔗 FileReader로 텍스트 파일 읽기

```
c:\temp\phone.txt를 출력합니다.
황기태 010-5555-7777
이재문 011-3333-4444
김남윤 065-2222-1111
```

힌트 FileReader를 이용하여 한 문자씩 읽어 출력하고자 하면 예제 8-1을 참고하고, 파일을 한 줄씩 읽고자 하면, Open Challenge의 힌트를 이용하거나 BufferedReader의 readLine() 메소드를 이용하면 된다.

3. c:\windows\system.ini 파일을 읽어 소문자를 모두 대문자로 바꾸어 출력하라.

🔗 텍스트 파일 읽기

난이도 4

힌트 이 문제는 여러 가지로 풀 수 있지만 대문자로 바꾸는 방법은 다음과 같다.

```
c = Character.toUpperCase(c); // c가 문자인 경우
String line;
line = line.toUpperCase(); // 문자열 line의 경우
```

4. c:\windows\system.ini 파일에 라인 번호를 붙여 출력하라. 난이도 5

🔗 텍스트 파일 읽기.
Scanner와 FileReader의 연결 연습

```
c:\windows\system.ini 파일을 읽어 출력합니다.
1: ; for 16-bit app support
2: [386Enh]
3: woafont=dosapp.fon
4: EGA80WOA.FON=EGA80WOA.FON
5: EGA40WOA.FON=EGA40WOA.FON
6: CGA80WOA.FON=CGA80WOA.FON
7: CGA40WOA.FON=CGA40WOA.FON
8:
9: [drivers]
10: wave=mmdrv.dll
11: timer=timer.driv
12:
13: [mci]
```

힌트

파일 입력 스트림과 Scanner를 이용하면 텍스트 파일을 라인 단위로 쉽게 읽을 수 있다.

```
Scanner fScanner = new Scanner(new FileReader("c:\\windows\\system.ini"));
while(scanner.hasNext()) { // 파일에 읽을 내용이 있는 동안
    String line = fScanner.nextLine(); // 파일에서 한 라인 읽기
```

또한 라인 번호를 다음과 같이 칸 맞춰 출력할 수 있다.

```
System.out.printf("%4d", lineNumber); // lineNumber는 라인 번호를 가진 int 변수
```

**🔗 FileInputStream과
FileOutputStream 활용 연습**

5. 2개의 파일을 입력받고 비교하여 같으면 "파일이 같습니다.", 틀리면 "파일이 서로 다릅니다."를 출력하는 프로그램을 작성하라. 텍스트 및 바이너리 파일 모두를 포함한다. 아래 실행 예시에서는 프로젝트 폴더에 elvis1.txt와 복사본 elvis1-복사본.txt를 미리 준비해 두었다. **난이도 5**

전체 경로명이 아닌 파일 이름만 입력하는 경우, 파일은 프로젝트 폴더에 있어야 합니다.
 첫번째 파일 이름을 입력하세요>>elvis1.txt
 두번째 파일 이름을 입력하세요>>elvis1-복사본.txt
 elvis1.txt와 elvis1-복사본.txt를 비교합니다.
 파일이 같습니다.

🔗 텍스트 파일 읽기와 쓰기

6. 사용자로부터 두 개의 텍스트 파일 이름을 입력받고 첫 번째 파일 뒤에 두 번째 파일을 덧붙인 새로운 파일을 생성하는 프로그램을 작성하라. 아래 실행 예시에서는 프로젝트 폴더에 elvis1.txt와 elvis2.txt를 미리 준비해 두었다. **난이도 5**

전체 경로명이 아닌 파일 이름만 입력하는 경우, 파일은 프로젝트 폴더에 있어야 합니다.
 첫번째 파일 이름을 입력하세요>>elvis1.txt
 두번째 파일 이름을 입력하세요>>elvis2.txt
 프로젝트 폴더 밑에 appended.txt 파일에 저장하였습니다.

🔗 바이너리 파일 복사 연습

7. 파일 복사 연습을 해보자. 이미지 복사가 진행되는 동안 10% 진행할 때마다 '*' 하나씩 출력하도록 하라. 아래 실행 예시에서는 프로젝트 폴더 밑에 a.jpg를 미리 준비해 두었다. **난이도 5**

a.jpg를 b.jpg로 복사합니다.
 10%마다 *를 출력합니다.

8. File 클래스를 이용하여 c:\에 있는 파일 중에서 제일 큰 파일의 이름과 크기를 출력하라. 난이도 5

🔗 File 클래스로 파일 리스트, 파일 크기 알아내기

가장 큰 파일은 c:\hiberfil.sys 3394002944바이트

9. c:\temp에 있는 .txt 파일만 삭제하는 프로그램을 작성하라. c:\나 c:\Windows 등의 디렉터리에 적용하면 중요한 파일이 삭제될 수 있으니 조심하라. 난이도 6

🔗 File 클래스로 파일 리스트, 파일 타입, 파일 삭제 연습

c:\temp디렉터리의 .txt 파일을 모두 삭제합니다....
 c:\temp\p.txt 삭제
 c:\temp\phone.txt 삭제
 c:\temp\test.txt 삭제
 총 3개의 .txt 파일을 삭제하였습니다.

힌트

문자열 s가 파일명을 담고 있는 경우 파일명에 .txt가 있는지 확인하려면, `int index = s.lastIndexOf(".txt")`를 이용하고 index가 -1이면 .txt 파일이 아닌 경우이다. 파일을 삭제하려면 다음과 같이 File 클래스의 `delete()` 메소드를 호출하면 된다.

```
File f = new File("c:\\temp\\p.txt");
f.delete(); // "c:\\temp\\p.txt" 파일 삭제
```

10. 전화번호를 미리 c:\temp\phone.txt 파일에 여러 개 저장해둔다. 이 파일을 읽어 다음 실행 예시와 같은 작동하는 검색 프로그램을 작성하라. 난이도 6

🔗 HashMap 컬렉션, 파일 입출력 실전 응용

총 7개의 전화번호를 읽었습니다.

이름>> 최박사

010-2222-3333

이름>> 백점만

011-1100-1100

이름>> 이상아

찾는 이름이 없습니다.

이름>> 이상형

010-3333-4444

이름>> 그만

최박사, 백점만, 이상형 학생을 포함하여 총 7명을 미리 다음과 같이 phone.txt 파일에 저장해 두었다.

최박사 010-2222-3333


백점만 011-1100-1100

이상형 010-3333-4444

...


힌트

이름을 '키'를 하고, 전화번호를 '값'으로 하는 `HashMap<String, String>`을 이용하고, 한 라인씩 읽어 이름과 전화번호를 해시맵에 저장하고, 사용자의 이름을 입력받아 검색하면 된다.

 Vector 컬렉션, 파일 입출력 실전 응용

11. words.txt 파일(출판사 홈페이지에 자료로 제공. 현재 이 문제의 정답 프로젝트에도 있음)에는 한 라인에 하나의 영어 단어가 들어 있다. 이 파일을 한 라인씩 읽어 Vector<String>에 라인별로 삽입하여 저장하고, 영어 단어를 입력받아 그 단어로 시작하는 모든 단어를 벡터에서 찾아 출력하는 프로그램을 작성하라. **난이도 7**

```
프로젝트 폴더 밑의 words.txt 파일을 읽었습니다...
단어>>lov
love
lovebird
lovelorn
단어>>kitt
kitten
kittenish
kittle
kitty
단어>>ajlfasf
발견할 수 없음
단어>>그만
종료합니다...
```

 Vector 컬렉션, 파일 입출력 실전 응용

12. 텍스트 파일에 있는 단어를 검색하는 프로그램을 작성해보자. 실행 예시는 프로젝트 폴더 밑에 자신이 작성한 자바 파일(예: test.java)을 복사하여 두고 읽은 경우이다. **난이도 8**

```
전체 경로명이 아닌 파일 이름만 입력하는 경우, 파일은 프로젝트 폴더에 있어야 합니다.
대상 파일명 입력>> test.java
검색할 단어나 문장>> void
4: public static void main(String[] args) {
검색할 단어나 문장>> int
9:   System.out.println("전체 경로명이 아닌 파일 이름만 입력하는 경우, 파일은 프로젝트 폴
  데에 있어야 합니다.");
11:   System.out.print("첫번째 파일 이름을 입력하세요>>");
14:   System.out.print("두번째 파일 이름을 입력하세요>>");
17:   System.out.println(src + "와 " + dst + "를 비교합니다.");
22:   System.out.println("파일이 같습니다.");
24:   System.out.println("파일이 다릅니다.");
29:   System.out.println("입출력 오류가 발생했습니다.");
39:   int srcCount=0, destCount;
49:   for (int i=0; i<srcCount; i++) {
검색할 단어나 문장>> for (
49:   for (int i=0; i<srcCount; i++) {
검색할 단어나 문장>> 그만
프로그램을 종료합니다.
```

13. 간단한 파일 탐색기를 만들어보자. 처음 시작은 c:\에서부터 시작한다. 명령은 크게 2가지로서 ".."를 입력하면 부모 디렉터리로 이동하고, "디렉터리명"을 입력하면 서브 디렉터리로 이동하여 파일리스트를 보여준다. **난이도 8**

***** 파일 탐색기입니다. *****

[c:\]

```
dir      12288바이트      Program Files
dir      12288바이트      Program Files (x86)
dir      8192바이트       ProgramData
```

..... 생략하였습니다.

```
dir      4096바이트       Users
dir      24576바이트      Windows
dir      8192바이트       Windows10Upgrade
dir      4096바이트       황기태
```

>>Windows

c:\Windows로 이동 지시

[c:\Windows]

```
dir      4096바이트       AppPatch
dir      0바이트          AppReadiness
dir      4096바이트       assembly
file      0바이트          ativpsrm.bin
```

..... 생략하였습니다.

```
dir      0바이트          System
file      219바이트       system.ini
dir      1048576바이트     System32
dir      12288바이트       SystemApps
dir      0바이트          Web
file      167바이트       win.ini
```

>>Web

c:\Windows\Web으로 이동 지시

[c:\Windows\Web]

```
dir      0바이트          4K
dir      0바이트          Screen
dir      0바이트          Wallpaper
```

>>..

c:\Windows로 이동 지시

[c:\Windows]

```
dir      4096바이트       AppPatch
dir      0바이트          AppReadiness
dir      4096바이트       assembly
file      0바이트          ativpsrm.bin
```

..... 생략하였습니다.

```
dir      0바이트          System
file      219바이트       system.ini
```


```

dir      1048576바이트   System32
dir      12288바이트    SystemApps
dir      0바이트        Web
file     167바이트      win.ini
>>그만

```

힌트

예제 8-8을 참고하고 File 클래스의 listFiles() 메소드를 활용하면 어렵지 않게 작성할 수 있다.

 File 클래스 활용

14. 문제 13을 확장하여 다음 명령을 추가하라. 난이도 7

```

>>rename phone.txt p.txt // phone.txt를 p.txt로 변경. 파일과 디렉터리 이름 변경
>>mkdir XXX // 현재 디렉터리 밑에 XXX 디렉터리 생성

```

***** 파일 탐색기입니다. *****

[c:\]

```

dir      12288바이트    Program Files
dir      12288바이트    Program Files (x86)
dir      8192바이트     ProgramData

```

..... 생략하였습니다

```

dir      4096바이트     Users
dir      24576바이트    Windows
dir      8192바이트     Windows10Upgrade
dir      4096바이트     황기태

```

>>mkdir 이재문

이재문 디렉터리를 생성하였습니다.

[c:\]

```

dir      12288바이트    Program Files
dir      12288바이트    Program Files (x86)
dir      8192바이트     ProgramData

```

..... 생략하였습니다

```

dir      4096바이트     Users
dir      24576바이트    Windows
dir      8192바이트     Windows10Upgrade
dir      0바이트        이재문
dir      4096바이트     황기태

```

생성된 이재문 디렉터리


```
>>temp
  [c:\temp]
dir      0바이트      HncDownload
file     66바이트     phone.txt
file     10바이트     test.txt
>>rename phone.txt
두 개의 파일명이 주어지지 않았습니다. !
>>rename phone.txt p.txt
phone.txt를 p.txt 이름 변경하였습니다.
  [c:\temp]
dir      0바이트      HncDownload
file     66바이트     p.txt
file     10바이트     test.txt
>>
```

phone.txt가 p.txt로 변경

힌트

rename phone.txt p.txt와 같은 명령을 읽고 문자열을 분리하기 위해 StringTokenizer를 이용하라. 만일 rename phone.txt 이런 식으로 명령을 잘못 입력한 경우 StringTokenizer.nextToken()은 NoSuchElementException 예외를 발생시키므로 try-catch 블록을 만들어 catch 블록에서 사용자에게 오류메시지를 출력하면 된다.