



16장. 파일 스트림

이것이 자바다 (<http://cafe.naver.com/thisjava>)

자바의 입출력 스트림

❖ 자바의 입출력 스트림

■ 입출력 장치와 자바 응용 프로그램 연결

- 입력 스트림 : 입력 장치로부터 자바 프로그램으로 데이터를 전달하는 객체
- 출력 스트림 : 자바 프로그램에서 출력 장치로 데이터를 보내는 객체

■ 특징

- 입출력 스트림 기본 단위 : 바이트
- 단방향 스트림, 선입선출 구조

자바 프로그램 개발자는 직접 입력 장치에서 읽지 않고 입력 스트림을 통해 읽으며, 스크린 등 출력 장치에 직접 출력하지 않고 출력 스트림에 출력하면 된다.



자바의 입출력 스트림 종류

■ 문자 스트림

- 2byte 유니코드 문자만 입출력하는 스트림
- 문자가 아닌 바이너리 데이터는 스트림에서 처리하지 못함
- 문자가 아닌 데이터를 문자 스트림으로 출력하면 깨진 기호가 출력
- 바이너리 파일을 문자 스트림으로 읽으면 읽을 수 없는 바이트가 생겨서 오류 발생

예) 텍스트 파일을 읽는 입력 스트림

■ 바이트 스트림

- 입출력 데이터를 단순 바이트의 흐름으로 처리
- 문자 데이터든 바이너리 데이터든 상관없이 처리 가능

예) 바이너리 파일을 읽는 입력 스트림

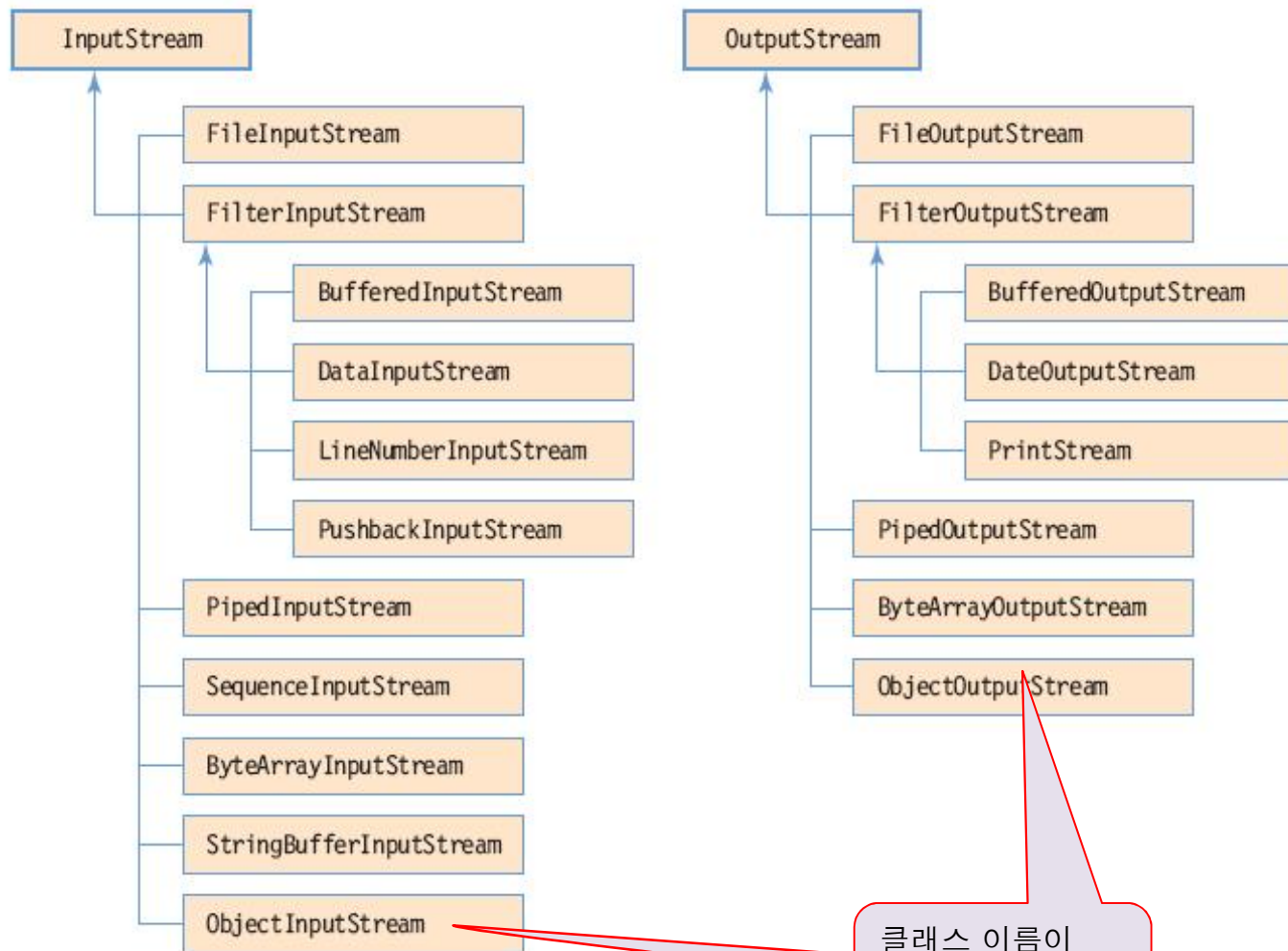


자바의 입출력 스트림 종류



JDK의 바이트 스트림 클래스 계층 구조

5

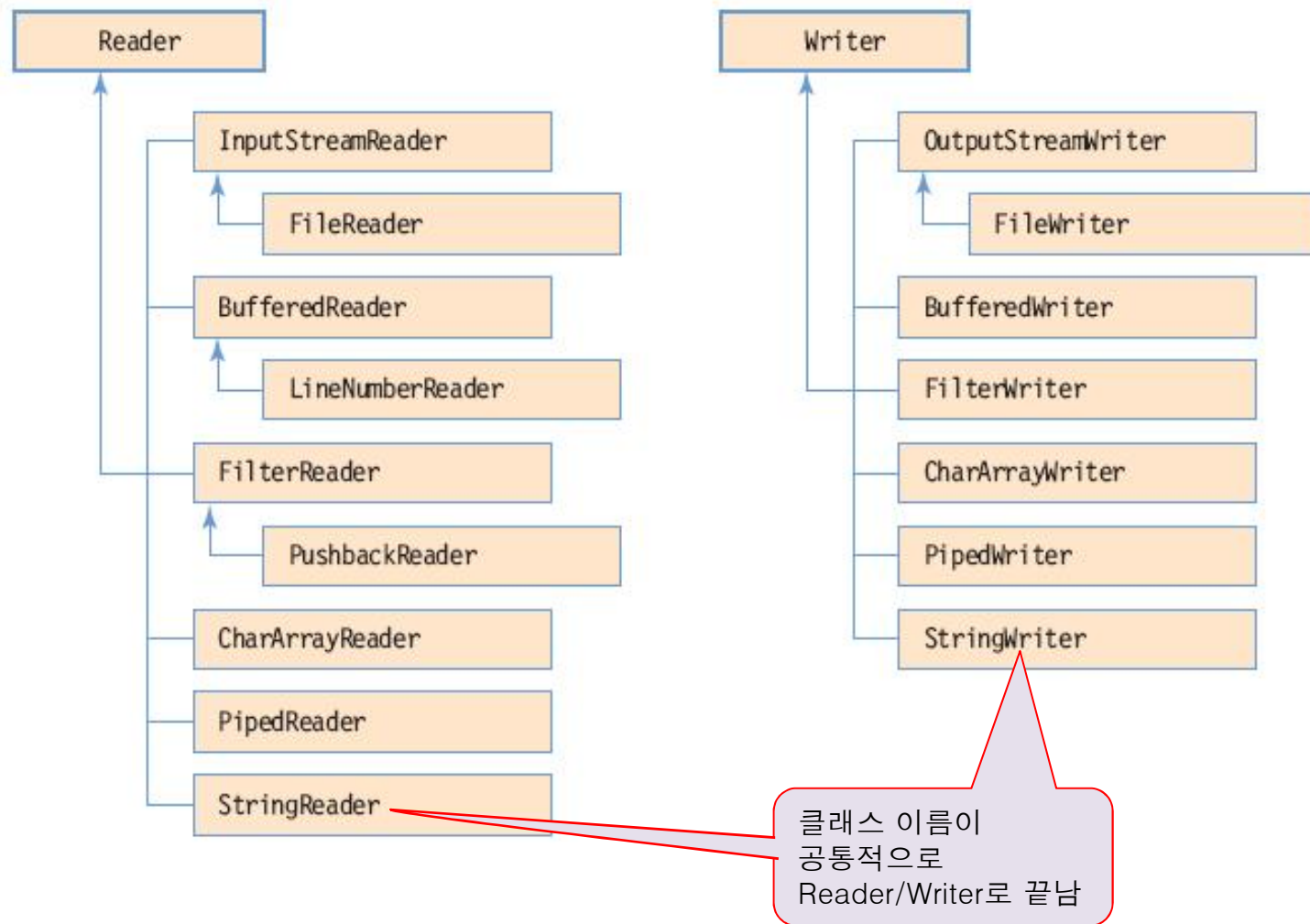


클래스 이름이
공통적으로
Stream으로 끝남



JDK의 문자 스트림 클래스 계층 구조

6



자바에서 파일I/O 코드 구조

7

❖ 1. 파일 스트림 생성

- `xxxstream = new xxxstream();`

- `InputStreamReader`
- `OutputStreamReader`
- `FileReader` class
- `FileWrite` class
- `FileInputStream` class
- `FileOutputStream` class

❖ 2. read, write

- `xxxstream .read()` or `xxxstream .write()`

❖ 3. 스트림 닫기

- `xxxstream .close()`



문자 스트림으로 텍스트 파일 읽기

8

❖ 텍스트 파일을 읽기 위해 문자 스트림 FileReader 클래스 이용

1. 파일 입력 스트림 생성(파일 열기)

- 스트림을 생성하고 파일을 열어 스트림과 연결

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

파일입력스트림

2. 파일 읽기

- read()로 문자 하나 씩 파일에서 읽음

```
int c;  
while((c = fin.read()) != -1) { // 문자를 c에 읽음. 파일 끝까지 반복  
    System.out.print((char)c); // 문자 c 화면에 출력  
}
```

read()는 파일의 끝(EOF)를
만나면 -1리턴

3. 스트림 닫기

- 스트림이 더 이상 필요 없으면 닫아야 함. 닫힌 스트림에서는 읽을 수 없음
- close()로 스트림 닫기

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



파일 입출력과 예외 처리

9

❖ 파일 입출력 동안 예외 발생 가능

■ 스트림 생성 동안 : **FileNotFoundException** 발생 가능

- 파일의 경로명이 틀리거나, 디스크의 고장 등으로 파일을 열 수 없음

```
FileReader fin = new FileReader("c:WWtest.txt"); // FileNotFoundException 발생가능
```

■ 파일 읽기, 쓰기, 닫기를 하는 동안 : **IOException** 발생 가능

- 디스크 오동작, 파일이 중간에 깨진 경우, 디스크 공간이 모자라서 파일 입출력 불가

```
int c = fin.read(); // IOException 발생 가능
```

❖ try-catch 블록 반드시 필요

■ 자바 컴파일러의 강제 사항

생략 가능. FileNotFoundException은 IOException을 상속받기 때문에 아래의 catch 블록 하나만 있으면 됨

```
try {
    FileReader fin = new FileReader("c:WWtest.txt");
    ..
    int c = fin.read();
    ...
} catch(FileNotFoundException e) {
    System.out.println("파일을 열 수 없음");
} catch(IOException e) {
    System.out.println("입출력 오류");
}
finally{
    try{fin.close();}catch(IOException e){}
}
```



FileReader의 생성자와 주요 메소드

10

생성자	설명
<code>FileReader(File file)</code>	file에 지정된 파일로부터 읽는 FileReader 생성
<code>FileReader(String name)</code>	name 이름의 파일로부터 읽는 FileReader 생성

메소드	설명
<code>int read()</code>	한 개의 문자를 읽어 정수형으로 리턴
<code>int read(char[] cbuf)</code>	최대 cbuf 배열의 크기만큼 문자들을 읽어 cbuf 배열에 저장. 만일 읽는 도중 EOF를 만나면 실제 읽은 문자 개수 리턴
<code>int read(char[] cbuf, int off, int len)</code>	최대 len 크기만큼 읽어 cbuf 배열의 off부터 저장. 읽는 도중 EOF를 만나면 실제 읽은 문자 개수 리턴
<code>String getEncoding()</code>	스트림이 사용하는 문자 집합의 이름 리턴
<code>void close()</code>	입력 스트림을 닫고 관련된 시스템 자원 해제



예제 : FileReader로 텍스트 파일 읽기

11

FileReader를 이용하여 d:\Whones.txt 파일을 읽어 화면에 출력하는 프로그램을 작성하라.

```
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;

public class ReadTextChar {
    public static void main(String[] args){
        FileReader filereader = null;
        try{
            //파일 객체 생성
            File file = new File("d:\\Whones.txt");
            //입력 스트림 생성
            filereader = new FileReader(file);
            int singleCh = 0;
            while((singleCh = filereader.read()) != -1){
                System.out.print((char)singleCh);
            }
        }catch (FileNotFoundException e) {
            // TODO: handle exception
        }catch(IOException e){
            System.out.println(e);
        }
        finally{
            try{filereader.close();}catch(IOException e){}
        }
    }
}
```

강감찬 010-1234-5681
김유신 010-1234-5682
정약용 010-1234-5683

파일 끝을 만나면 -1 리턴



❖ student_info.txt

이름	학번	전화번호	학과
강감찬	2012685	010-1234-5681	경제학과
김유신	2013635	010-1234-5682	수의학과
정약용	2017654	010-1234-5683	통계학과

- String[] 에 데이터를 저장
- 메뉴 1. 이름으로 검색 -> 학번, 전화번호, 학과 출력
- 메뉴 2. 학번으로 검색 -> 이름, 전화번호
- 메뉴 3. 학과으로 검색 => 이름, 학번



❖ 한 반의 학생의 과목별 점수가 파일에 저장

subject_score.txt

이름	국어	영어	수학	과학	사회	음악	체육
김철수	80	60	70	100	80	90	100
박유철	60	90	90	100	90	90	70
김영희	100	100	60	70	90	90	100
최수지	80	40	50	70	80	100	100

❖ 1. 각 학생의 평균 점수를 출력하세요

- 이름 - 평균점수

❖ 2. 각 학생들의 석차를 출력하세요

- 이름 - 석차

❖ 3. 각 과목의 평균점수를 출력하고 , 최대 득점자를 출력하세요(복수면 복수명 출력)

- 과목 - 평균점수 - 이름



❖ 부서원의 급여를 파일에 저장

salary.txt

이름,실수령액,세금,국민연금

김철수,180,20,15

박유철,160,15,13

김영희,200,10,12

최수지,180,20,11

❖ 1. 부서원의 실수령액 를 출력하세요

- 이름 – 실수령액

❖ 2. 부서의 평균 실수령액 를 출력하세요

- 실수령액

- -----

❖ 3. 부서원의 본급여를 출력하세요

❖ 4. 세금+국민연금의 금액이 가장 작은 부서원의 이름과 금액을 출력하세요



실습

❖ 지출내용을 파일에 입력

expense.txt

날짜,내역,사용금액,각달총사용금액

```
20190215$아침식사 $ 6000 $ 480000
20190311 $ 점심식사 $ 10000 $ 260000
20190312$옷구입 $ 30000 $ 290000
20190315 $ 식료품구입 $ 50000 $ 310000
20190321 $ 저녁식사 $ 12000 $ 312000
```

- ❖ 1. 날짜별 사용금액을 출력하세요
 - 날짜 -사용금액
- ❖ 2. 3월 총 사용금액을 출력하세요
- ❖ 3. 3월에 사용한 총 식사비의 총액 및 평균금액을 구하세요
- ❖ 4. 3월지출중 가장 많은 금액의 내역을 출력하세요



Buffered Reader, Writer의 주요 메소드

메서드명	기능
BufferedReader(Reader rd)	rd에 연결되는 문자입력 버퍼스트림 생성
BufferedWriter(Writer wt)	wt에 연결되는 문자출력 버퍼스트림 생성
int read()	스트림으로부터 한 문자를 읽어서 int 형으로 리턴
int read(char[] buf)	문자배열 buf의 크기만큼 문자를 읽어들이. 읽어들이 문자 수를 리턴
int read(char[] buf, int offset, int length)	buf의 offset위치에서부터 length 길이만큼 문자를 스트림으로부터 읽어들이
String readLine()	스트림으로부터 한 줄을 읽어 문자열로 리턴
void mark()	현재우치를 마킹, 차 후 reset() 을 이용하여 마킹위치부터 시작함
void reset()	마킹이 있으면 그 위치에서부터 다시시작, 그렇지 않으면 처음부터 다시시작
long skip(int n)	n 개의 문자를 건너 뛴
void close()	스트림 닫음
void write(int c)	int 형으로 문자 데이터를 출력문자스트림으로 출력
void write(String s, int offset, int length)	문자열 s를 offset 위치부터 length 길이만큼을 출력스트림으로 출력
void write(char[] buf, int offset, int length)	문자배열 buf의 offset 위치부터 length 길이만큼을 출력스트림으로 출력
void newLine()	줄바꿈 문자열 출력
void flush()	남아있는 데이터를 모두 출력시킴.



예제 : BufferedReader로 텍스트 파일 읽기

```
17 import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;

public class ReadTextLine {
    public static void main(String[] args){
        FileReader filereader= null;
        File file = null;
        BufferedReader bufReader = null;
        try{
            //파일 객체 생성
            file = new File("d:WWphones.txt");
            //입력 스트림 생성
            filereader = new FileReader(file);
            //입력 버퍼 생성
            bufReader = new BufferedReader(filereader);
            String line = "";
            while((line = bufReader.readLine()) != null){
                System.out.println(line);
            }
        }catch (FileNotFoundException e) {
            // TODO: handle exception
        }catch(IOException e){
            System.out.println(e);
        }
        finally{
            try{bufReader .close();}catch(IOException e){}
        }
    }
}
```

강감찬 010-1234-5681
김유신 010-1234-5682
정약용 010-1234-5683



FileOutputStream의 생성자와 주요 메소드

18

생성자	설명
<code>FileOutputStream(File file)</code>	<code>file</code> 이 지정하는 파일에 출력하는 <code>FileOutputStream</code> 생성
<code>FileOutputStream(String name)</code>	<code>name</code> 이 지정하는 파일에 출력하는 <code>FileOutputStream</code> 생성
<code>FileOutputStream</code> <code>(File file, boolean append)</code>	<code>FileOutputStream</code> 을 생성하며 <code>append</code> 가 <code>true</code> 이면 <code>file</code> 이 지정하는 파일의 마지막부터 데이터 저장
<code>FileOutputStream</code> <code>(String name, boolean append)</code>	<code>FileOutputStream</code> 을 생성하며 <code>append</code> 가 <code>true</code> 이면 <code>name</code> 이름의 파일의 마지막부터 데이터 저장

메소드	설명
<code>void write(int b)</code>	<code>int</code> 형으로 넘겨진 한 바이트를 출력 스트림으로 출력
<code>void write(byte[] b)</code>	배열 <code>b</code> 의 바이트를 모두 출력 스트림으로 출력
<code>void write(byte[] b, int off, int len)</code>	<code>len</code> 크기만큼 <code>off</code> 부터 배열 <code>b</code> 를 출력 스트림으로 출력
<code>void flush()</code>	출력 스트림에서 남아 있는 데이터 모두 출력
<code>void close()</code>	출력 스트림을 닫고 관련된 시스템 자원 해제

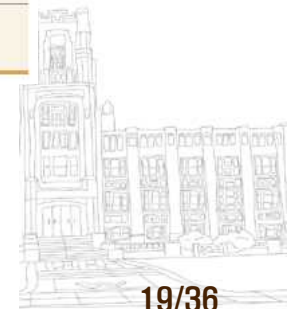


FileInputStream의 생성자와 주요 메소드

19

생성자	설명
<code>FileInputStream(File file)</code>	file이 지정하는 파일로부터 읽는 FileInputStream 생성
<code>FileInputStream(String name)</code>	name이 지정하는 파일로부터 읽는 FileInputStream 생성

메소드	설명
<code>int read()</code>	입력 스트림에서 한 바이트를 읽어 int형으로 리턴
<code>int read(byte[] b)</code>	최대 배열 b의 크기만큼 바이트를 읽음. 읽는 도중 EOF를 만나면 실제 읽은 바이트 수 리턴
<code>int read(byte[] b, int off, int len)</code>	최대 len개의 바이트를 읽어 b 배열의 off 위치에 저장. 읽는 도중 EOF를 만나면 실제 읽은 바이트 수 리턴
<code>int available()</code>	입력 스트림에서 현재 읽을 수 있는 바이트 수 리턴
<code>void close()</code>	입력 스트림을 닫고 관련된 시스템 자원 해제



예제 : BufferedWriter로 텍스트 파일 쓰기

```
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;

public class WriteTextLine {

    public static void main(String[] args){
        File file = null;
        BufferedWriter bufferedWriter = null;
        try{
            //파일 객체 생성
            file = new File("d:\\new.txt");
            bufferedWriter = new BufferedWriter(new FileWriter(file));

            if(file.isFile() && file.canWrite()){
                //쓰기
                bufferedWriter.write("문자열 추가1");
                //개행문자쓰기
                bufferedWriter.newLine();
                bufferedWriter.write("문자열 추가2");
            }
        }catch (IOException e) {
            System.out.println(e);
        }
        finally{
            try{bufferedWriter .close();}catch(IOException e){}
        }
    }
}
```

new.txt

문자열 추가1
문자열 추가2

❖ 아래의 배열의 값을 파일에 쉼표로 구분하여 출력하세요

- `String[] strArr={ “apple” , “orange” ,
melon” ,” mango” };`
- 파일내용
 - apple, orange, melon, mango

❖ 아래의 배열의 값을 파일에 빈칸으로 구분하여 출력하세요

- `int[] intArr={1,2,3,4,5,6,7,8,7,9};`
- 파일 내용
 - 첫번째 라인 - 짝수 인덱스 의 값들
 - 두번째 라인 - 홀수 인덱스 의 값들
 - 세번째 라인 - 합과 평균값



❖ student_info.txt

이름	학번	전화번호	학과
강감찬	2012685	010-1234-5681	경제학과
김유신	2013635	010-1234-5682	수의학과
정약용	2017654	010-1234-5683	통계학과

- 다음의 결과를 파일에 저장하세요
- 메뉴 1. 이름으로 검색 -> 학번, 전화번호, 학과 출력
- 메뉴 2. 학번으로 검색 -> 이름, 전화번호
- 메뉴 3. 학과으로 검색 => 이름, 학번

저장형식

```
메뉴1
XXXXXXXXX
메뉴2
XXXXXXXXX
메뉴3
XXXXXXXXX
```



실습

❖ 한 반의 학생의 과목별 점수가 파일에 저장

subject_score.txt

이름	국어	영어	수학	과학	사회	음악	체육
김철수	80	60	70	100	80	90	100
박유철	60	90	90	100	90	90	70
김영희	100	100	60	70	90	90	100
최수지	80	40	50	70	80	100	100

❖ 다음의 결과를 파일에 저장하세요

❖ 1. 각 학생의 평균 점수를 출력하세요

- 이름 - 평균점수

❖ 2. 각 학생들의 석차를 출력하세요

- 이름 - 석차

❖ 3. 각 과목의 평균점수를 출력하고, 최대 득점자를 출력하세요(복수면 출력)

- 과목 - 평균점수 - 이름



실습

❖ 부서원의 급여를 파일에 저장

salary.txt

이름,실수령액,세금,국민연금

김철수,180,20,15

박유철,160,15,13

김영희,200,10,12

최수지,180,20,11

❖ 다음의 결과를 파일에 저장하세요

❖ 1. 부서원의 실수령액 를 출력하세요

- 이름 - 실수령액

❖ 2. 부서의 평균 실수령액 를 출력하세요

- 실수령액

- -----

❖ 3. 부서원의 본급여를 출력하세요

❖ 4. 세금+국민연금의 금액이 가장 작은 부서원의 이름과 금액을 출력하세요



실습

❖ 지출내용을 파일에 입력

expense.txt

날짜,내역,사용금액,각달총사용금액

```
20190215$아침식사 $ 6000 $ 480000
20190311 $ 점심식사 $ 10000 $ 260000
20190312$옷구입 $ 30000 $ 290000
20190315 $ 식료품구입 $ 50000 $ 310000
20190321 $ 저녁식사 $ 12000 $ 312000
```

❖ 다음의 결과를 파일에 저장하세요

❖ 1. 날짜별 사용금액을 출력하세요

■ 날짜 - 사용금액

❖ 2. 3월 총 사용금액을 출력하세요

❖ 3. 3월에 사용한 총 식사비의 총액 및 평균금액을 구하세요

❖ 4. 3월지출중 가장 많은 금액의 내역을 출력하세요



예제 : 파일 복사

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

public class CopyTextFile {
    public static void main(String[] args){
        BufferedReader bufReader= null;
        BufferedWriter bufferedWriter = null;
        try{
            //파일 객체 생성
            File file = new File("d:WWphones.txt");
            //입력 스트림 생성
            FileReader filereader = new FileReader(file);
            //입력 버퍼 생성
            bufReader = new BufferedReader(filereader);
            String line = "";
            //파일 객체 생성
            File wfile = new File("d:WWnew.txt");
            bufferedWriter = new BufferedWriter(new FileWriter(wfile));
```



예제 : 파일 복사

```
while((line = bufReader.readLine()) != null){
    System.out.println(line);
    bufferedWriter.write(line);
    //개행 문자쓰기
    bufferedWriter.newLine();
}

}catch (FileNotFoundException e) {
    // TODO: handle exception
}catch(IOException e){
    System.out.println(e);
}

        finally{
            try{bufReader .close();}catch(IOException e){}
            try{bufferedWriter .close();}catch(IOException e){}
        }
    }
}
```



예제 : 버퍼 스트림 바이너리 파일 복사

28

```
import java.io.*;

public class BufferedBinaryFileCopy {
    public static void main(String[] args) {
        File src = new File("c:\\Users\\Public\\Pictures\\Sample Pictures\\desert.jpg");
        File dst = new File("d:\\dest.jpg"); // 목적 파일
        FileInputStream fi = null;
        FileOutputStream fo = null;
        BufferedInputStream in = null;
        BufferedOutputStream out = null;
        int c;
        try {
            fi = new FileInputStream(src);
            fo = new FileOutputStream(dst);
            in = new BufferedInputStream(fi);
            out = new BufferedOutputStream(fo);
            while ((c = in.read()) != -1) {
                out.write((char)c);
            }
        } catch (IOException e) {
            System.out.println("파일 복사 오류");
        }
        finally{
            try{in.close();}catch(IOException e){}
            try{out.close();}catch(IOException e){}
        }
    }
}
```

