객체지향프로그래밍응용 과제04 컴퓨터소프트웨어공학과 2-YA 20202296 전채린

[예제 8-1]

[소스코드]

// 예제 8-1

// FileReader로 텍스트 파일 읽기

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) {

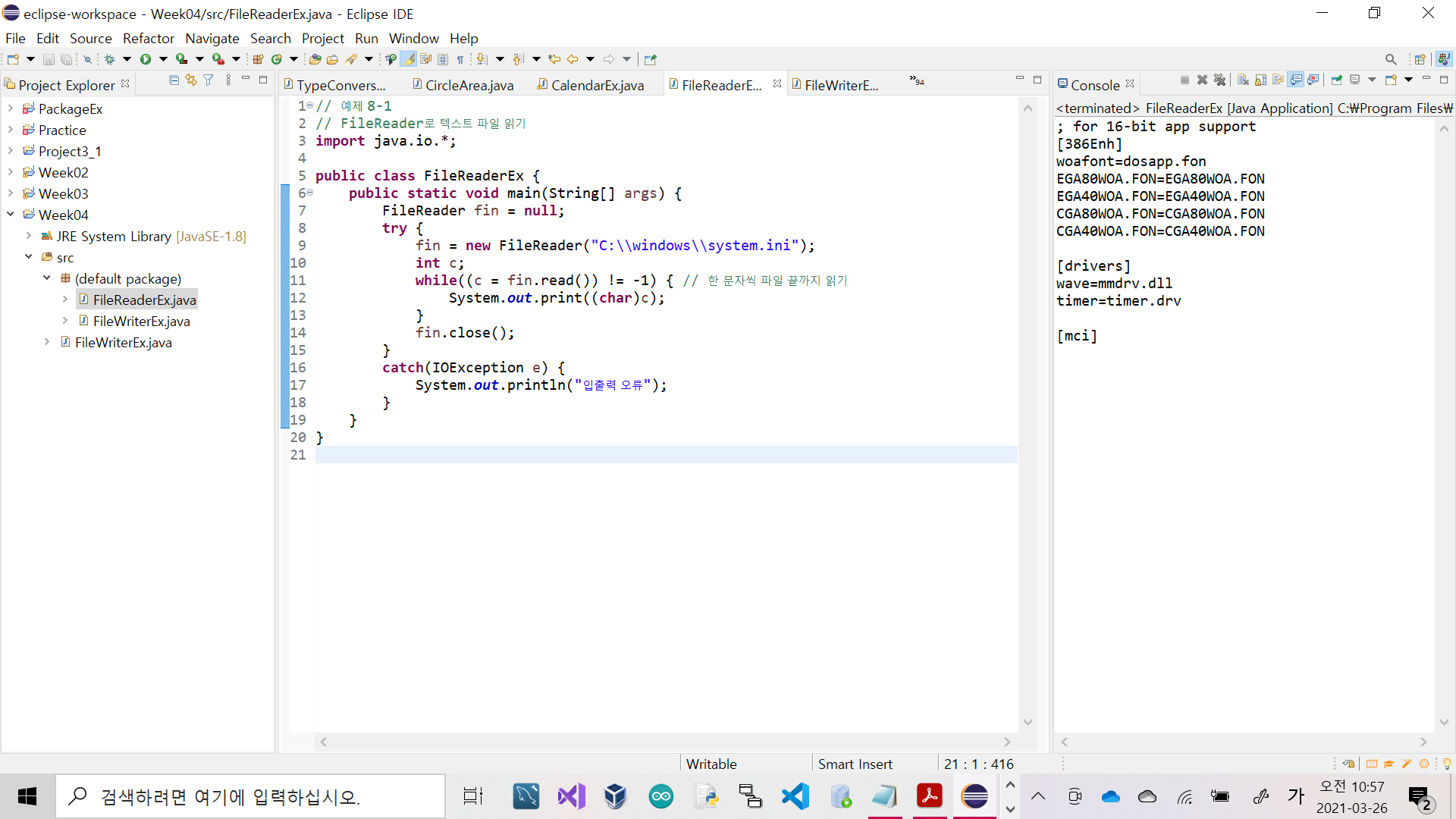
System.out.println("입출력 오류");

}

}

}

[실행결과]



[예제 8-4]

[소스코드]

// 예제 8-4

// 키보드 입력을 파일로 저장하기

// Scanner를 이용하여 키보드에서 입력받은 데이터를 c:\Temp\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("c:\\Temp\\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");

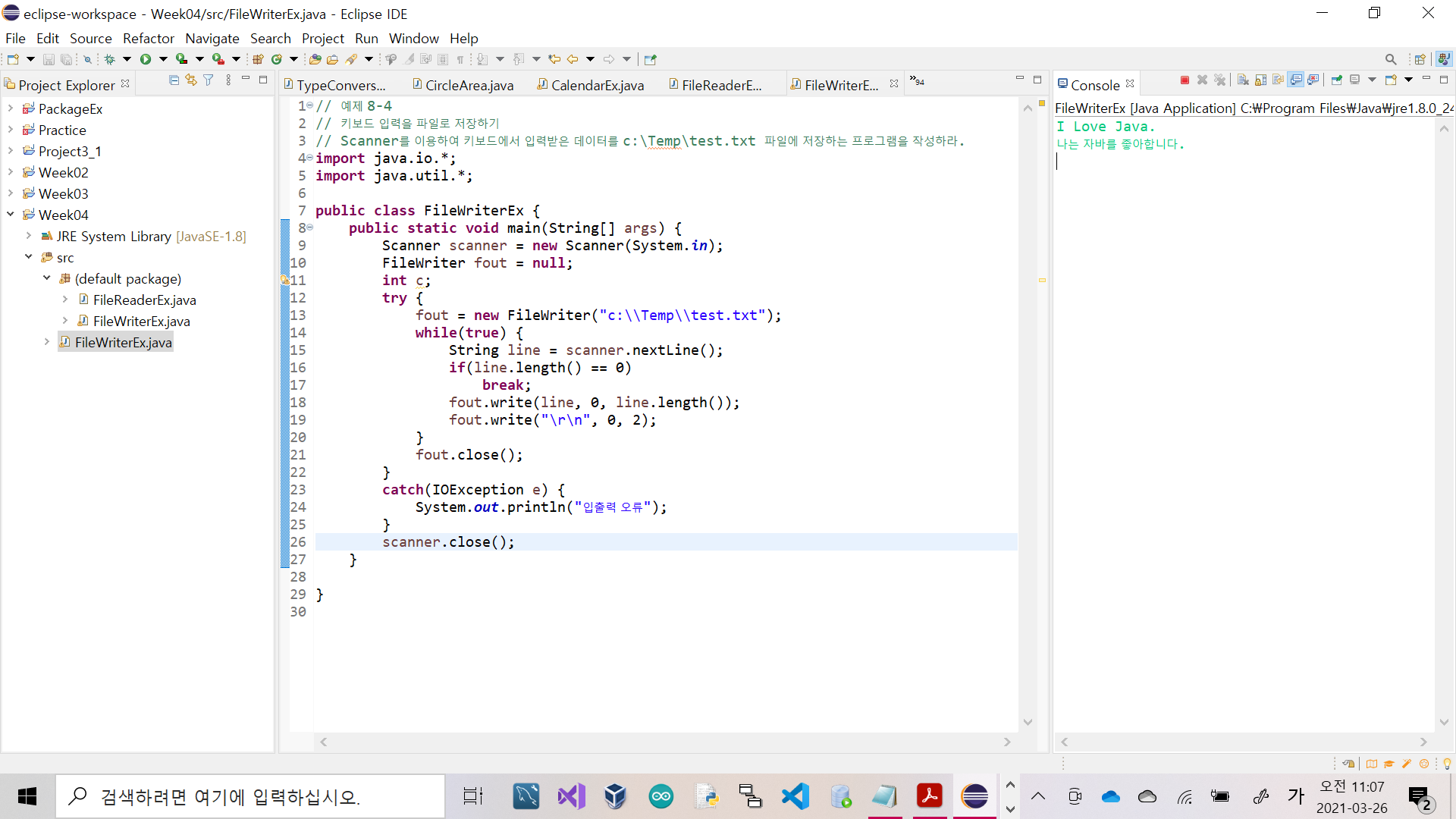
}

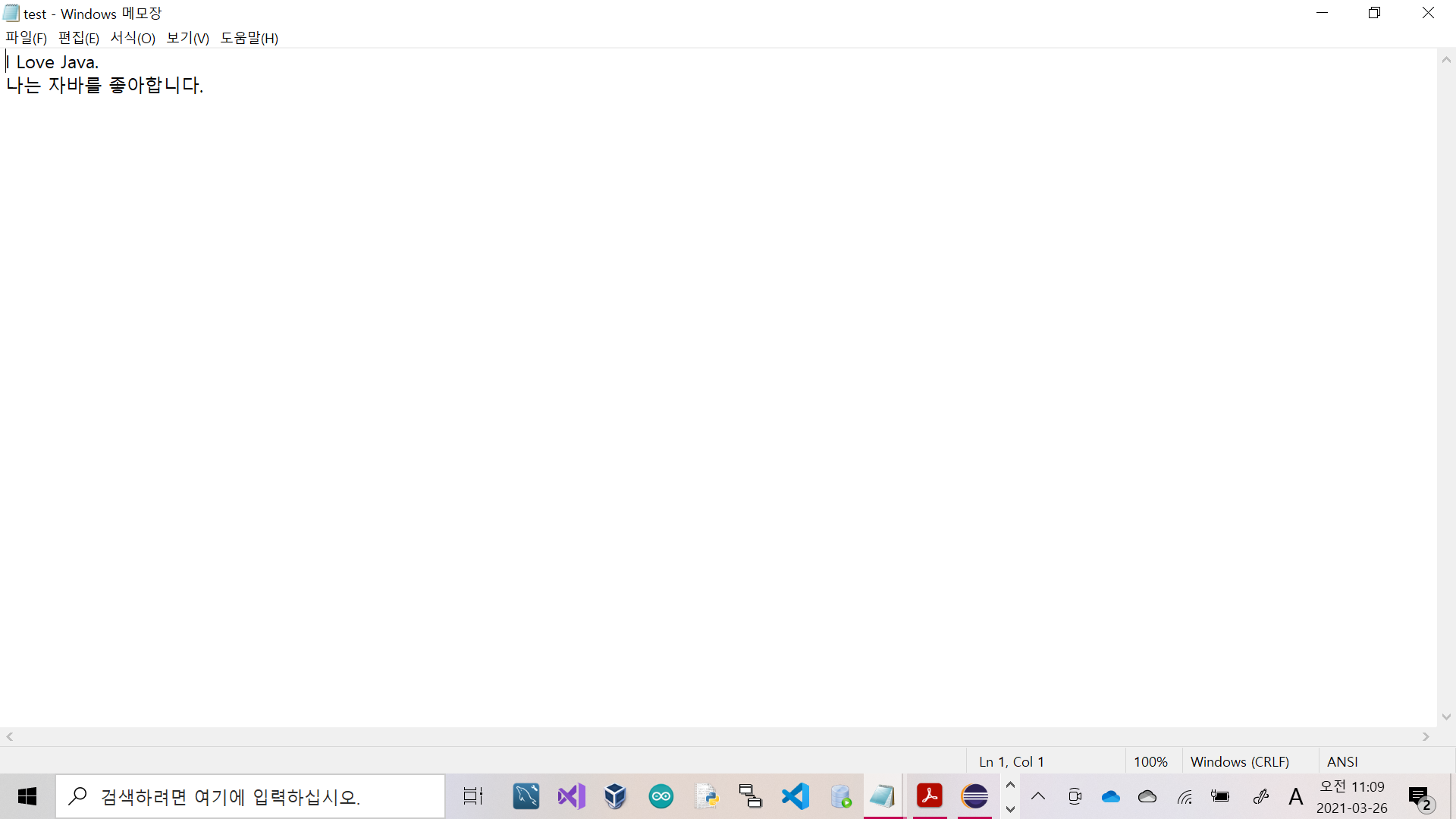
scanner.close();

}

}

[실행결과]





[예제 \*\_\* CopyFile]

[소스코드]

// 예제 \*\_\* CopyFile

// 소스 파일을 읽어 파일로 복사하고 콘솔에 출력하기

import java.io.\*;

public class CopyFile {

public static void main(String[] args) {

String input = "c:\\Temp\\dream.txt";

String output = "c:\\Temp\\dreamCopy.txt";

try {

FileReader fin = new FileReader(input);

FileWriter fout = new FileWriter(output);

int ch;

while((ch = fin.read()) != -1) { // 한 문자씩 파일 끝까지 읽기

fout.write((char)ch);

System.out.print((char)ch);

}

fin.close();

fout.close();

}

catch(Exception e) {

e.printStackTrace();

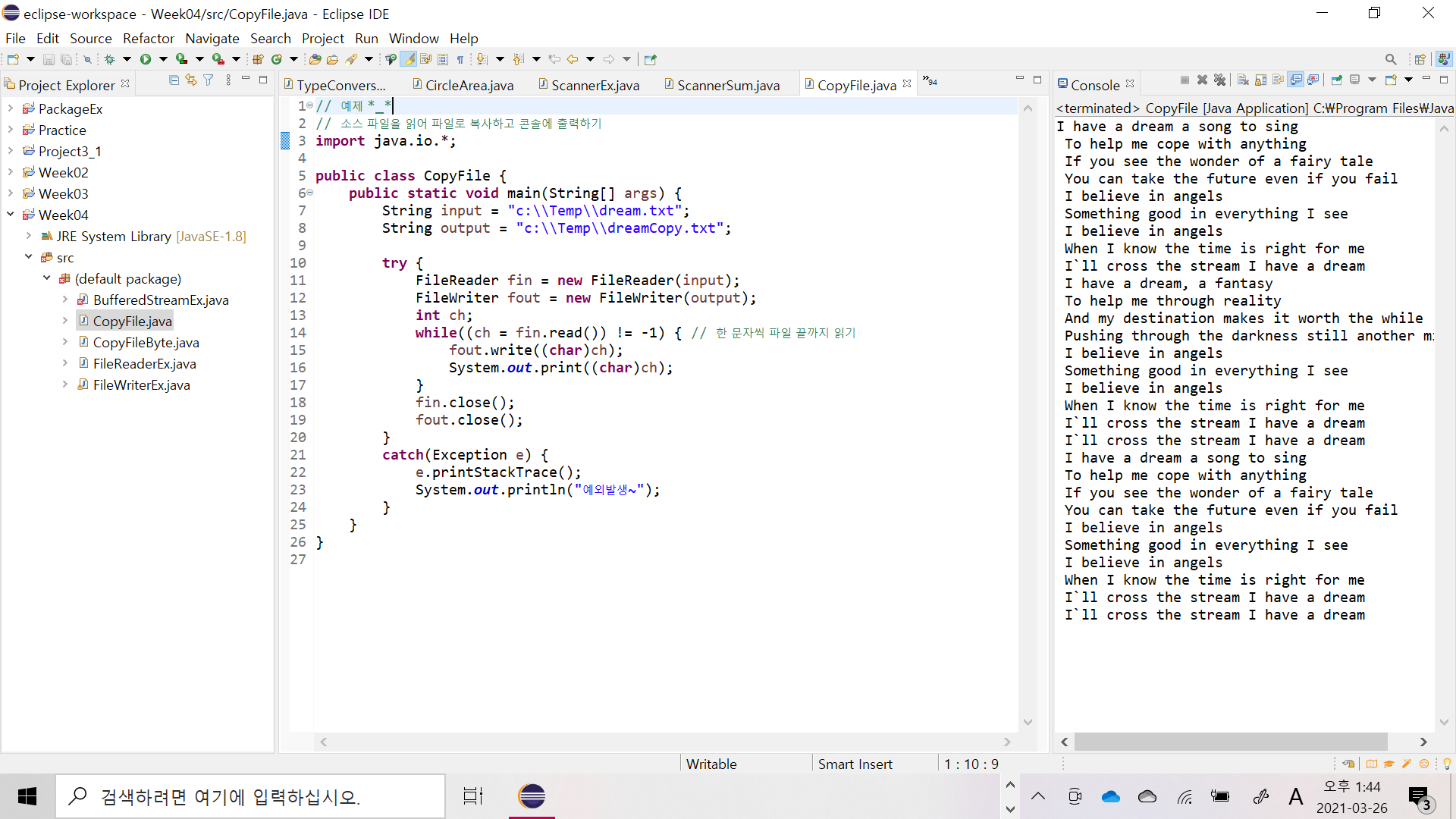
System.out.println("예외발생~");

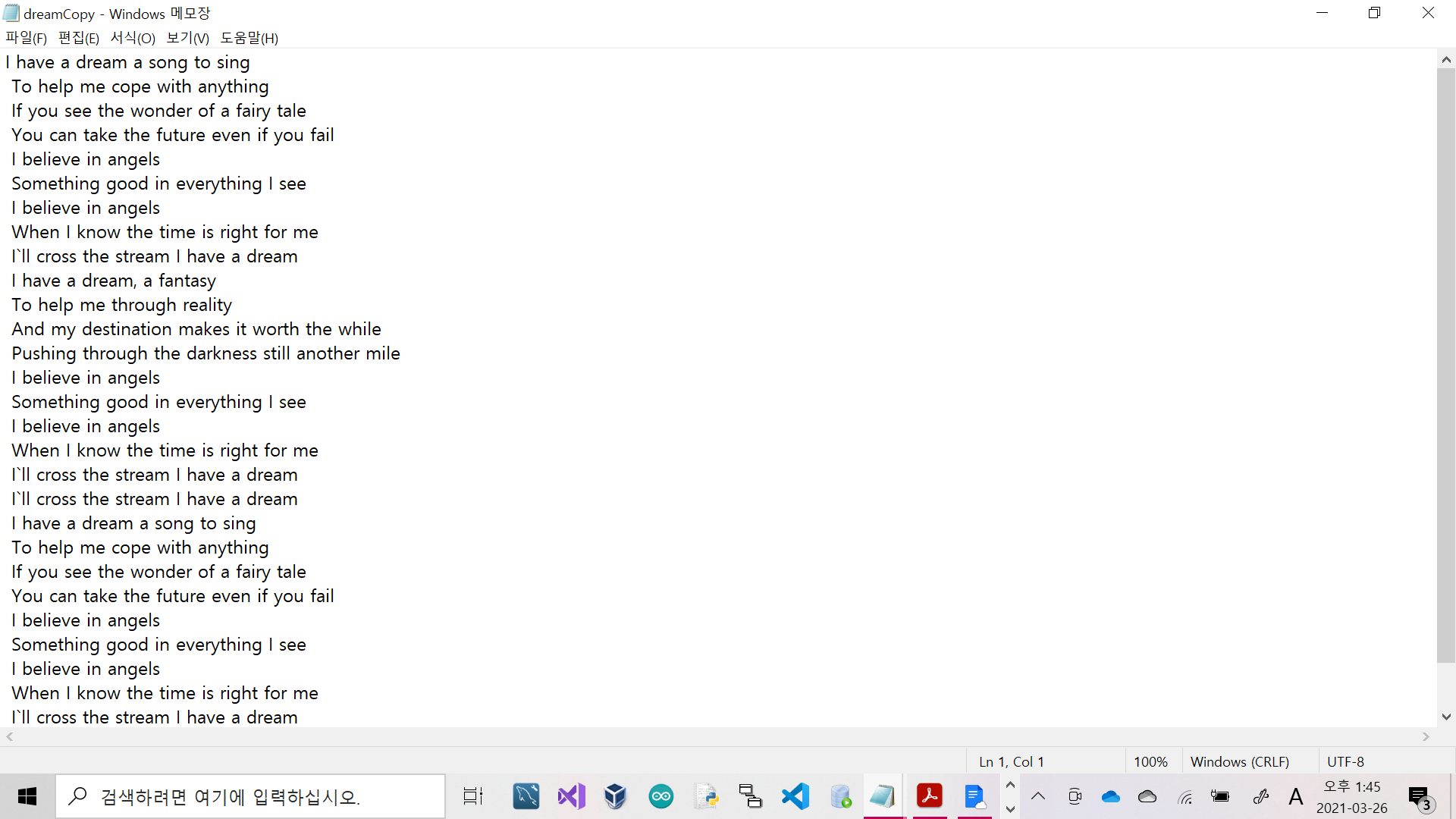
}

}

}

[실행결과]





[예제 CopyFileByte]

[소스코드]

// CopyFileByte

// 파일 복사 예제

import java.io.\*;

public class CopyFileByte {

public static void main(String[] args) {

String input = "c:\\Temp\\bts.mp3";

String output = "c:\\Temp\\btsCopy.mp3";

try {

FileInputStream fin = new FileInputStream(input);

FileOutputStream fout = new FileOutputStream(output);

int ch;

while((ch = fin.read()) != -1) { // 한 바이트씩 파일 끝까지 읽기

fout.write(ch);

}

System.out.println("작업완료했습니다.");

fin.close();

fout.close();

}

catch(Exception e) {

e.printStackTrace();

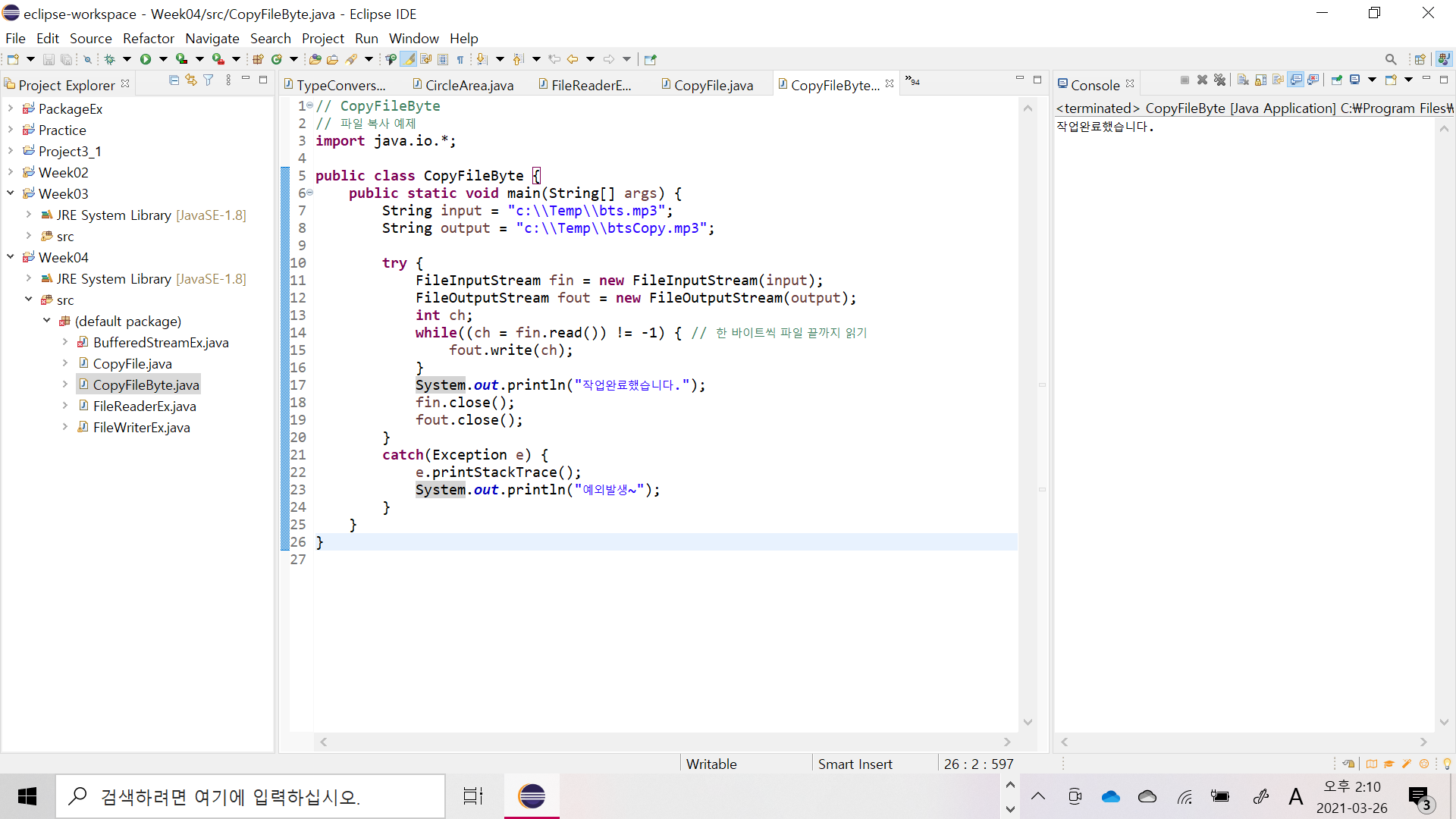
System.out.println("예외발생~");

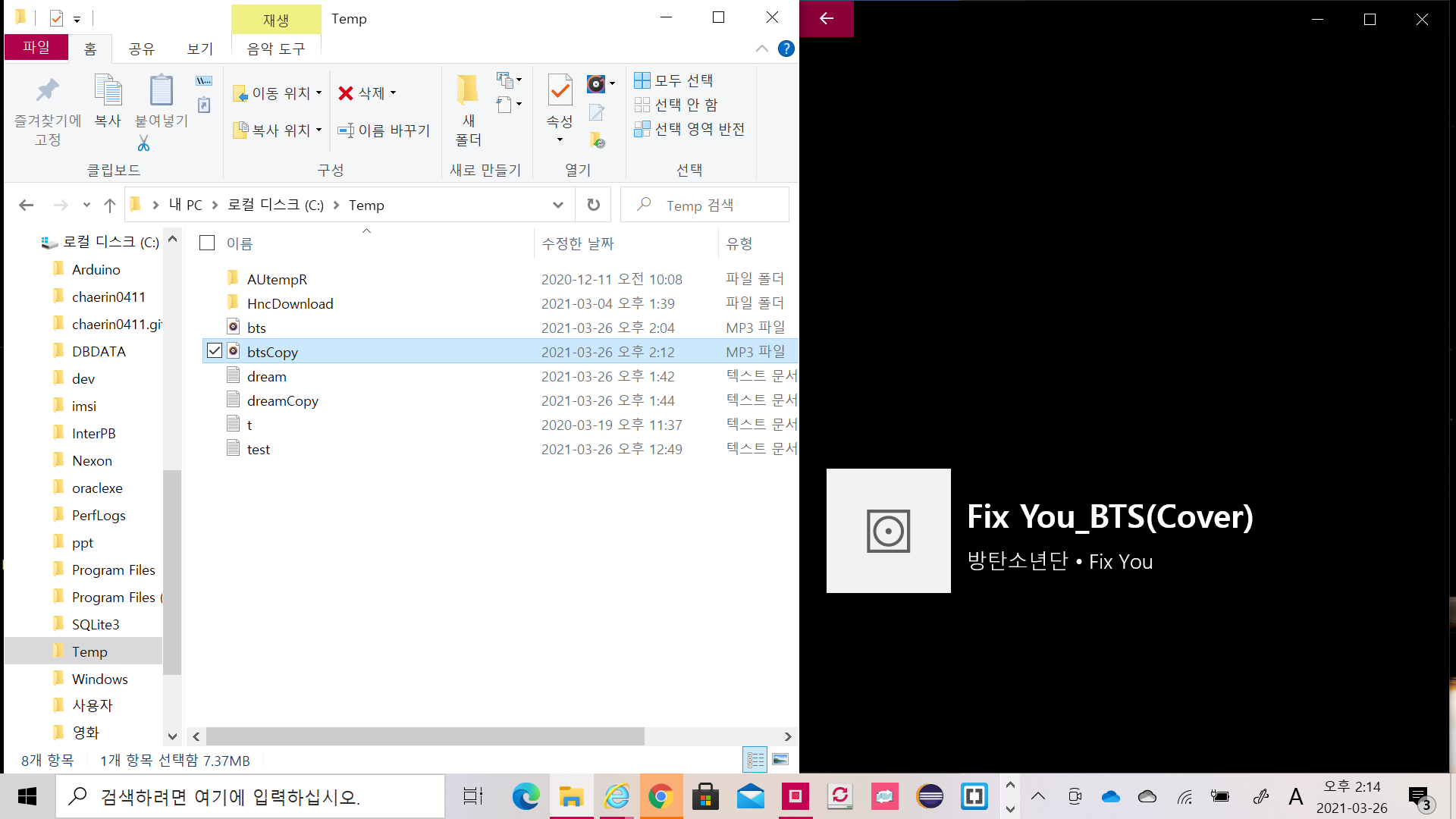
}

}

}

[실행결과]





[예제 BufferedStreamEx]

[소스코드]

// 예제 BufferedStreamEx

// 버퍼 스트림을 이용한 출력

// 버퍼 사용 유무 비교

import java.io.\*;

public class BufferedStreamEx {

public static void main(String[] args) {

long start, end, duration;

String input = "c:\\Temp\\dream.txt";

String output = "c:\\Temp\\dreamcopy.txt";

// 버퍼 사용 복사

start = System.nanoTime();

try {

FileInputStream fin = new FileInputStream(input);

FileOutputStream fout = new FileOutputStream(output);

BufferedInputStream bin = new BufferedInputStream(fin);

BufferedOutputStream bout = new BufferedOutputStream(fout);

int ch;

while((ch = bin.read()) != -1) { // 한 문자씩 파일 끝까지 읽기

bout.write(ch);

}

System.out.println("버퍼 작업이 완료되었습니다.");

fin.close();

fout.close();

}

catch(Exception e) {

e.printStackTrace();

System.out.println("버퍼 사용 예외발생~");

}

end = System.nanoTime();

duration = end - start;

System.out.println("버퍼를 사용한 경우 : " + duration);

// 일반 파일 복사

start = System.nanoTime();

try {

FileInputStream fin = new FileInputStream(input);

FileOutputStream fout = new FileOutputStream(output);

int ch;

while((ch = fin.read()) != -1) { // 한 문자씩 파일 끝까지 읽기

fout.write(ch);

}

System.out.println("파일 작업이 완료되었습니다.");

fin.close();

fout.close();

}

catch(Exception e) {

e.printStackTrace();

System.out.println("예외발생~");

}

end = System.nanoTime();

duration = end - start;

System.out.println("파일만 사용한 경우 : " + duration);

}

}

[실행결과]

