

2025년 상반기 K-디지털 트레이닝

데이터 입출력

[KB] IT's Your Life



- ☑ 다음 변수의 값을 "c:/temp/test1.db" 파일에 저장하세요.
 - 스트림은 기본 스트림을 사용함

```
package ch18.sec02.exam01;

public class WriteExample {
          public static void main(String[] args) {
                byte a = 10;
                byte b = 20;
                byte c = 30;
          }
}
```

WriteExample.java

```
package ch18.sec02.exam01;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;
public class WriteExample {
         public static void main(String[] args) {
                   try(OutputStream os = new FileOutputStream("C:/temp/test1.db")) {
                            byte a = 10;
                            byte b = 20;
                            byte c = 30;
                            os.write(a);
                            os.write(b);
                            os.write(c);
                   } catch (IOException e) {
                            e.printStackTrace();
```

- 다음 배열의 값을 "c:/temp/test2.db"에 저장하세요.
 - 스트림은 기본 스트림을 사용함

```
package ch18.sec02.exam02;

public class WriteExample {
         public static void main(String[] args) {
               byte[] array = { 10, 20, 30 };
         }
}
```

WriteExample.java

```
package ch18.sec02.exam02;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;
public class WriteExample {
         public static void main(String[] args) {
                   try(OutputStream os = new FileOutputStream("C:/Temp/test2.db")) {
                            byte[] array = \{ 10, 20, 30 \};
                            os.write(array);
                   } catch (IOException e) {
                            e.printStackTrace();
```

- ☑ "c:/temp/test1.db"에서 바이트 데이터를 읽어 화면에 출력하세요.
 - 스트림은 기본 스트림을 사용함

```
package ch18.sec03.exam01;

public class ReadExample {
     public static void main(String[] args) {
     }
}
```

ReadExample.java

```
package ch18.sec03.exam01;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;
public class ReadExample {
         public static void main(String[] args) {
                  try (InputStream is = new FileInputStream("C:/Temp/test1.db")){
                           while(true) {
                                     int data = is.read();
                                                                // 1byte씩 읽기
                                     if(data == -1) break;
                                                                          // 파일 끝에 도달했을 경우
                                     System.out.println(data);
                  } catch (FileNotFoundException e) {
                           e.printStackTrace();
                  } catch (IOException e) {
                           e.printStackTrace();
```

- ☑ "c:/temp/test2.db"에서 바이트 데이터를 읽어 바이트 배열(data)에 저장하세요.
 - 스트림은 기본 스트림을 사용함
 - 1바이트 씩 읽어, 화면에 출력함

```
package ch18.sec03.exam02;
public class ReadExample {
        public static void main(String[] args) {
            byte[] data;
        }
}
```

ReadExample.java

```
package ch18.sec03.exam02;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;
public class ReadExample {
public static void main(String[] args) {
 try(InputStream is =
             new FileInputStream("C:/Temp/test2.db")) {
                           byte[] data = new byte[100];
                           while(true) {
                           int num = is.read(data);
                           // 최대 100 byte 읽기
                           if(num == -1) break;
                                                      // 파일 끝 도달
                           for(int i=0; i<num; i++) {
                                        System.out.println(data[i]);
} catch (FileNotFoundException e) {
                                         e.printStackTrace();
                           } catch (IOException e) {
                                         e.printStackTrace();
```

☑ test.jpg 파일을 준비하고, 이 파일을 test2.jpg 파일로 복사하세요

스트림은 기본 스트림을 사용함

☑ CopyExample.java

```
package ch18.sec03.exam03;
public class CopyExample {
          public static void main(String[] args) throws Exception {
                    String originalFileName = "C:/Temp/test.jpg";
                    String targetFileName = "C:/Temp/test2.jpg";
                   try(
                              InputStream is = new FileInputStream(originalFileName);
                              OutputStream os = new FileOutputStream(targetFileName);
                    ) {
                              byte[] data = new byte[1024]; // 배열 버퍼 생성
                             while (true) {
                                        int num = is.read(data);
                                                                               // 최대 1024바이트 읽기
                                        if (num == -1) break;
                                                                               // 파일을 다 읽으면 while 문 종료
                                        os.write(data, 0, num);
                                                                               // 읽은 데이터 파일에 쓰기
                             os.flush();
                                                 // 내부 버퍼 잔류 바이트를 출력하고 버퍼를 비움
```

☑ CopyExample.java

☑ 다음 변수의 값들을 c:/temp/test.txt에 저장하세요.

```
package ch18.sec04.exam01;

public class WriteExample {
    public static void main(String[] args) {
        char a = 'A';
        char b = 'B';
        char[] arr = { 'C', 'D', 'E' };
}
```

WriteExample.java

```
package ch18.sec04.exam01;
import java.io.FileWriter;
import java.io.IOException;
import java.io.Writer;
public class WriteExample {
         public static void main(String[] args) {
                   //문자 기반 출력 스트림 생성
                   try(Writer writer = new FileWriter("C:/Temp/test.txt")) {
                            //1 문자씩 출력
                            char a = 'A';
                            writer.write(a);
                            char b = 'B';
                            writer.write(b);
                            //char 배열 출력
                            char[] arr = { 'C', 'D', 'E' };
                            writer.write(arr);
```

☑ CopyExample.java

```
//문자열 출력
writer.write("FGH");

//버퍼에 잔류하고 있는 문자들을 출력하고, 버퍼를 비움
writer.flush();

} catch (IOException e) {
    e.printStackTrace();
}

}
```

- ☑ "c:/temp/test.txt"에 저장된 텍스트를 읽어 화면에 출력하세요.
 - 스트림은 기본 스트림을 사용함
 - 한 번은 한 글자씩 처리
 - 한 번은 글자 배열로 처리

```
package ch18.sec04.exam02;

public class ReadExample {
     public static void main(String[] args) {
     }
}
```

ReadExample.java

```
package ch18.sec04.exam02;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.io.Reader;
public class ReadExample {
         public static void main(String[] args) {
                   try {
                            Reader reader = null;
                            //1 문자씩 읽기
                            reader = new FileReader("C:/Temp/test.txt");
                            while(true) {
                                      int data = reader.read();
                                      if(data == -1) break;
                                      System.out.print((char)data);
                            reader.close();
                            System.out.println();
```

ReadExample.java

```
//문자 배열로 읽기
         reader = new FileReader("C:/Temp/test.txt");
         char[] data = new char[100];
         while(true) {
                  int num = reader.read(data);
                  if(num == -1) break;
                  for(int i=0; i<num; i++) {
                            System.out.print(data[i]);
         reader.close();
} catch (FileNotFoundException e) {
         e.printStackTrace();
} catch (IOException e) {
         e.printStackTrace();
```

- ♡ 다음 문자열을 "c:/temp/test.txt"에 저장하고, 그 파일에서 읽어서 화면에 출력하세요.
 - 문자 변환 보조 스트림을 사용함
 - 문자열은 UTF-8 문자셋으로 저장하고, 읽어 옴

```
package ch18.sec06;
public class CharacterConvertStreamExample {
    public static void main(String[] args) {
        String str = "문자 변환 스트림을 사용합니다.";
    }
}
```

CharacterConvertStreamExample.java

```
package ch18.sec06;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.io.Reader;
import java.io.Writer;
public class CharacterConvertStreamExample {
         public static void main(String[] args) throws Exception {
                  String str = "문자 변환 스트림을 사용합니다.";
                  write(str);
                  String data = read();
                  System.out.println(data);
```

CharacterConvertStreamExample.java

```
public static void write(String str) throws Exception {
         OutputStream os = new FileOutputStream("C:/Temp/test.txt");
         Writer writer = new OutputStreamWriter(os, "UTF-8");
         writer.write(str);
         writer.flush();
         writer.close();
public static String read() throws Exception {
         InputStream is = new FileInputStream("C:/Temp/test.txt");
         Reader reader = new InputStreamReader(is, "UTF-8");
         char[] data = new char[100];
         int num = reader.read(data);
         reader.close();
         String str = new String(data, 0, num);
         return str;
```

- ☑ c:/temp/test3.txt 파일을 라인 단위로 읽어, 화면에 한 줄씩 출력하세요.
 - o test3.txt 파일에 여러 줄의 텍스트를 입력해 둠
 - 스트림은 성능 향상 보조 스트림을 사용함

ReadLineExample.java

```
package ch18.sec07.exam02;
import java.io.*;
public class ReadLineExample {
          public static void main(String[] args) throws Exception {
                   BufferedReader br = new BufferedReader(
                             new FileReader("src/ch18/sec07/exam02/ReadLineExample.java")
                   );
                   int lineNo = 1;
                   while(true) {
                             String str = br.readLine();
                             if(str == null) break;
                             System.out.println(lineNo + "\text{\text{W}}t" + str);
                             lineNo++;
                   br.close();
```

💟 다음 조건을 처리하는 코드를 작성하세요.

- C:/temp/file1.txt이 존재하는지 확인하세요.
- 존재하는 경우 파일인지, 디렉토리인지 구분함
 - 파일이면 파일경로, 파일 크기를 출력하세요.
 - 디렉토리이면 "<dir> 파일 경로" 형태로 출력하세요.
- 존재하지 않는 경우, "해당 파일은 없는 파일입니다." 출력함

```
package ch18.sec11;

public class FileExample {
         public static void main(String[] args) {
         }
}
```

FileExample.java

```
package ch18.sec11;
import java.io.File;
import java.text.SimpleDateFormat;
import java.util.Date;
public class FileExample {
         public static void main(String[] args) throws Exception {
                   //File 객체 생성
                   File dir = new File("C:/Temp/images");
                   File file1 = new File("C:/Temp/file1.txt");
                   File file2 = new File("C:/Temp/file2.txt");
                   File file3 = new File("C:/Temp/file3.txt");
                   //존재하지 않으면 디렉토리 또는 파일 생성
                   if(dir.exists() == false) { dir.mkdirs(); }
                   if(file1.exists() == false) { file1.createNewFile(); }
                   if(file2.exists() == false) { file2.createNewFile(); }
                   if(file3.exists() == false) { file3.createNewFile(); }
```

FileExample.java

```
//Temp 폴더의 내용을 출력
File temp = new File("C:/Temp");
File[] contents = temp.listFiles();
SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd a HH:mm");
for(File file : contents) {
          System.out.printf("%-25s", sdf.format(new Date(file.lastModified())));
          if(file.isDirectory()) {
                    System.out.printf("%-10s%-20s", "<DIR>", file.getName());
          } else {
                    System.out.printf("%-10s%-20s", file.length(), file.getName());
          System.out.println();
                                           2023-05-17 오후 16:15
                                                                 <DIR>
                                                                         2020
                                           2024-01-19 오후 16:52
                                                                 3683
                                                                        chapter06.sql
                                           2024-01-22 오전 11:08
                                                                       file1.txt
                                           2024-01-22 오전 11:08
                                                                       file2.txt
                                           2024-01-22 오전 11:08
                                                                       file3.txt
                                                                 0
                                           2024-01-10 오후 13:21
                                                                 415
                                                                        Hello.class
                                           2024-01-10 오후 13:18
                                                                        Hello.java
                                                                 108
                                           2024-01-22 오전 11:08
                                                                 <DIR>
                                                                        images
                                           2024-01-22 오전 11:04
                                                                 201
                                                                        object.dat
```