Java프로그래밍

9강. java.io 패키지와 스트림 (교재 8장)

컴퓨터과학과 김희천 교수

한국방송통신대학교

오늘의 학습목차

1. 스트림
2. 바이트 스트림
3. 캐릭터 스트림
4. 파일 입출력
5. 콘솔 입출력과 보조 스트림

1. 스트림

1) Java 언어와 스트림

- Java 언어에서 스트림을 통해 입출력을 수행할 수 있음
  - ✓ 입력 스트림은 데이터 생산자(소스)와 연결
  - ✓ 출력 스트림은 데이터 소비자(목적지)와 연결
- 다양한 입출력 종류(디스크, 문자 배열, 네트워크 소켓, 다른 프로그램 등)에 상관없이 동일한 방법으로 프로그램을 작성
  - ✓ 스트림을 통해 입출력을 제어함

스트림

- 순서가 있는 일련의 데이터 흐름을 의미
- 데이터 생산자(소스)와 데이터 소비자(목적지) 사이의 데이터가 지나는 통로

2) 스트림의 분류(1)

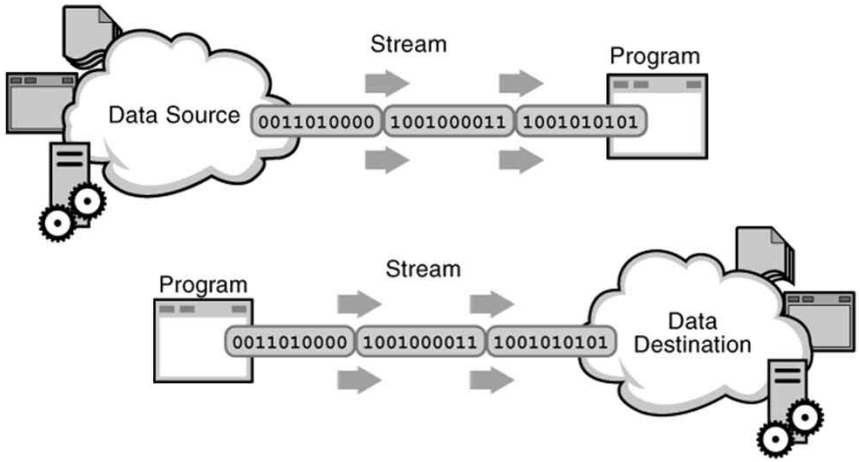
◆ 입력 스트림과 출력 스트림

✓ 프로그램은 입력 스트림으로부터 데이터를 읽을 수 있음

- 데이터 소스가 설정되어야 함

✓ 프로그램은 출력 스트림으로 데이터를 쓸 수 있음

- 데이터 목적지가 설정되어야 함



2) 스트림의 분류(2)

◆ 바이트 스트림과 캐릭터 스트림

✓ 바이트 스트림은 byte 단위로 데이터를 다룸

- xxxInputStream과 xxxOutputStream

✓ 캐릭터 스트림은 char 단위로 데이터를 다룸

- xxxReader과 xxxWriter

◆ 기본 스트림과 보조 스트림

✓ 기본 스트림은 입출력 기능을 제공하는 스트림

✓ 보조 스트림은 자체적으로 입출력 기능을 수행할 수는 없어서 기본 스트림과 함께 사용되어야 하며, 보조 기능을 제공하는 스트림

3) 스트림 관련 클래스

◆ java.io 패키지의 스트림 클래스

처리 단위 \ 처리 방향		입력 스트림	출력 스트림
기본 스트림	바이트 스트림	InputStream FileInputStream	OutpuStream FileOutputStream
	문자 스트림	Reader FileReader	Writer FileWriter
보조 스트림	바이트 스트림	BufferedInputStream DataInputStream ObjectInputStream	BufferedOutputStream PrintStream DataOutputStream ObjectOutputStram
	문자 스트림	BufferedReader	BufferedWriter PrintWriter
	기타	InputStreamReader	OutputStreamWriter

4) 스트림 사용하기

◆ 어떤 스트림 클래스를 사용할 것인가

✓ 입력 스트림? 출력 스트림?

✓ 데이터 생산자와 소비자를 결정

- 기본 스트림을 반드시 사용해야 함

✓ 문자 단위? 바이트 단위?

✓ 보조 스트림이 필요한가?

- 필요하다면 기본 스트림 객체를 먼저 생성하고,
  기본 스트림을 감싸 보조 스트림 객체를 생성함

```
FileInputStream fis = new FileInputStream( );
BufferedInputStream bis = new BufferedInputStream(fis);
```

2. 바이트 스트림

2. 바이트 스트림

1) InputStream 클래스

◆ 바이트 단위 입력 스트림 클래스의 최상위 클래스

주요 메소드

◆ abstract int read( )
  - ✓ 입력 스트림으로부터 1 바이트를 읽어 정수로 리턴함

◆ int read(byte[ ] b)
  - ✓ 입력 스트림으로부터 읽어서 byte 배열에 저장
  - ✓ 읽어 들인 바이트 개수를 리턴함

◆ int read(byte[ ] b, int off, int len)

◆ int available( )
  - ✓ 다음 read( )할 때, 블로킹 없이 입력 스트림에서 읽을 수 있는 데이터 길이

◆ long skip(long n)
  - ✓ 입력 스트림에서 n 바이트를 건너 뜀

◆ void mark(int readlimit), void reset( )

2) OutputStream 클래스

- 바이트 단위 출력 스트림 클래스의 최상위 클래스
- 추상 클래스이며 이것의 하위 클래스는 xxxOutputStream

주요 메소드

- void write(int b)
  - ✓ 1 바이트의 데이터를 출력 스트림에 씀
- void write(byte[ ] b)
  - ✓ byte 배열의 내용을 출력 스트림에 씀
- void write(byte[ ] b, int off, int len)
- void close( )
- void flush( )

3. 캐릭터 스트림

3. 캐릭터 스트림

1) Reader 클래스

- 입력용 캐릭터 단위 스트림 클래스의 최상위 클래스
- 추상 클래스이며 이것의 하위 클래스는 xxxReader

주요 메소드

- int read( )
  - ✓ 1개 문자(2 바이트)를 읽어 리턴함
- int read(char[ ] cbuf)
  - ✓ 문자를 읽어 char 배열에 저장함
  - ✓ 읽어 들인 문자의 개수를 리턴함
- boolean ready( )
  - ✓ 스트림이 읽힐 준비가 되었으면 true를 리턴함
- abstract void close( )
  - ✓ 입력 스트림을 닫고 자원을 반납함

2) Writer 클래스

- 출력용 캐릭터 단위 스트림 클래스의 최상위 클래스
- 추상 클래스이며 이것의 하위 클래스는 xxxWriter

주요 메소드

- void write(int c)
  - ✓ 1개의 문자(2 바이트)를 출력함
- void write(char[ ] cbuf)
- void write(String str)
- void write(String str, int off, int len)
- abstract void close( )

4. 파일 입출력

4. 파일 입출력

1) File 클래스

- 파일이나 디렉터리를 표현
  - ✓ 상대 또는 절대 경로를 가짐
- 파일/디렉터리를 조작할 수 있는 메소드 제공
  - ✓ 이름과 경로의 조회
  - ✓ 파일과 디렉토리의 생성과 삭제
    - 입출력 메소드는 제공되지 않음

생성자

- File(String pathname)
  - ✓ pathname은 상대 또는 절대 경로로 표현될 수 있음
  - ✓ File myFile = new File("c:₩₩temp₩₩data.txt");

2) File 클래스의 주요 메소드(1)

◆ 주요 메소드

- ✓ boolean exists( )
- ✓ boolean isDirectory( ), boolean isFile( )
- ✓ String getName( )
- ✓ String getPath( )
- ✓ long length( )
- ✓ boolean createNewFile( )
- ✓ boolean delete( )
- ✓ boolean mkdir( ), boolean mkdirs( )

2) File 클래스의 주요 메소드(2)

◆ 주요 메소드

✓ String[ ] list( ), File[ ] listFiles( )

- File 객체에 지정된 디렉터리 안에 들어 있는 파일과 서브 디렉터리들의 이름을 문자열 배열(또는 경로를 File 배열)로 반환

✓ String getParent( ), File getParentFile( )

- 상위 경로의 이름(또는 File 객체)을 반환

✓ static File[ ] listRoots( )

- 루트 디렉터리들을 File 배열로 반환

3) File 클래스 예제

```
import java.io.*;

public class ListDirectory2 {
   public static void main(String args[]) {
      File file = new File("c:\\windows");
      File files[ ] = file.listFiles( );
      int i = 0;
      while(i < files.length) {
          System.out.print(files[i].getPath( ));
          System.out.println("\t" + files[i].length( ));
          i++;
      }
   }
}
```

```
c:₩windows₩addins          0
c:₩windows₩afreeca.ico     353118
c:₩windows₩AhnInst.log     121513
```

4) RandomAccessFile 클래스

◆ 랜덤 엑세스 파일

✓ 파일의 임의 위치에서 읽기/쓰기가 가능

- 파일을 오픈하고, 위치를 지정하고, 읽기/쓰기를 함

✓ 읽고 쓰는 위치는 파일 포인터가 가리킴

- 읽기(또는 쓰기)는 파일 포인터가 가리키는 위치부터 바이트 단위로 읽음(또는 씀)

✓ byte 단위로 읽고 쓰며, 파일 포인터가 이동됨

- 랜덤 엑세스 파일을 커다란 byte 배열로 볼 수 있음

5) RandomAccessFile 클래스의 메소드

- 생성자
  - ✓ RandomAccessFile(File file, String mode),
  - ✓ RandomAccessFile(String name, String mode)
    - mode는 읽기 전용의 " r " 또는 읽기/쓰기 겸용의 "rw"

주요 메소드

- int read( )
- int read(byte[ ] b), int read(byte[ ] b, int off, int len)
- void seek(long pos)
- void write(byte[ ] b),
- void write(byte[ ] b, int off, int len)

6) RandomAccessFile 클래스 예제

```
import java.io.*;

public class RAFTest {
   public static void main(String args[ ]) {
       try {
            RandomAccessFile raf;
            raf = new RandomAccessFile("c:\\java\\test.txt", "rw");
            for( int i = 0; i < 10; i++)
              raf.write(i);
            for(int i = 9;i >= 0;i--) {
              raf.seek(i);
              System.out.print(raf.read( ));
            }
            raf.close();
       } catch (Exception e) { System.out.println(e); }
     }
}
```

```
9876543210
```

7) FileInputStream과 FileOutputStream 클래스

- ✓ 파일로부터 데이터를 읽기/쓰기 위한 입력/출력용 기본 스트림
- ✓ 바이트 단위의 입력/출력

◆ FileInputStream 클래스의 생성자
- ✓ 기존 파일과 연결된 입력 스트림 객체를 생성
- ✓ FileInputStream(File file), FileInputStream(String name)

◆ FileOutputStream 클래스의 생성자
- ✓ 기존 파일(없다면 생성)과 연결된 출력 스트림 객체를 생성
- ✓ FileOutputStream(String name)
- ✓ FileOutputStream(File file, boolean append)

8) FileOutputStream 예제

```
import java.io.*;
public class FileOutputStreamTest {
   public static void main(String args[ ]) {
     try {
        File inFile, outFile;
        inFile = new File("c:\\Java\\FileInputStreamTest.java");
        outFile = new File("c:\\Java\\FileTemp.java");
        InputStream is = new FileInputStream(inFile);
        OutputStream os = new FileOutputStream(outFile);
        int nData;
        nData = is.read( );
        while(nData != -1) {
              os.write(nData);
              nData = is.read( );
        }
        is.close( );  os.close( );
     } catch (Exception e) {
        System.out.println(e);
     }
   }
}
```

9) FileReader와 FileWriter 클래스

- 텍스트 파일을 다루기 위한 기본 스트림
- 문자 단위의 입력/출력

생성자

- FileReader(File file)
- FileReader(String fileName)

- FileWriter(File file)
- FileWriter(String fileName)
- FileWriter(File file, boolean append)

5. 콘솔 입출력과 보조 스트림

5. 콘솔 입출력과 보조 스트림

1) Console 클래스

- 콘솔 입출력을 제공하는 클래스
  - ✓ 키보드 입력과 화면 출력을 편리하게 지원
- 명령 프롬프트 창에서 실행해야 함
- System.console( )을 사용하여 콘솔 객체를 생성함

주요 메소드

- String readLine( )
  - ✓ 한 라인을 읽음
- char[ ] readPasssword( )
  - ✓ 입력할 때 화면에 보이지 않음
- PrintWriter writer( ), Reader reader( )

2) Console 클래스 예제

```
import java.io.*;

public class ConsoleTest {
   public static void main(String args[]) {
      String name;
      char[ ] pw;
      Console con = System.console( );

      System.out.print("name : ");
      name = con.readLine( );
      System.out.print("password : ");
      pw = con.readPassword( );

      PrintWriter pr = con.writer( );
      pr.println("name : " + name);
      pr.println("password : " + pw);
   }
}
```

```
>java ConsoleTest
name : kildong↲
password :     ↲
name : kildong
password : [C@16d3586
```

3) 보조 스트림

- 기본 스트림의 성능을 높이거나 보조 기능을 제공하는 스트림
- 입출력 기능을 직접 수행하지는 못함
- 보조 스트림을 생성할 때, 기본 스트림 객체를 생성자의 인자로 이용함
  - ✓ '기본 스트림을 보조 스트림으로 감싼다'라고 함
  - ✓ 프로그램에서는 보조 스트림을 사용해 입출력하면 됨

```
FileInputStream fis = new FileInputStream( );
BufferedInputStream bis = new BufferedInputStream(fis);
```

4) 보조 스트림의 종류

- 버퍼링 기능의 제공
  - ✓ BufferedInputStream, BufferedOutputStream
  - ✓ BufferedReader, BufferedWriter
- Java의 기본 자료형을 그대로 읽기/쓰기 위한 기능의 제공
  - ✓ DataInputStream, DataOutputStream
- 다양한 출력 형식의 제공
  - ✓ PrintStream, PrintWriter
- 텍스트 파일을 라인 단위로 읽는 메소드를 제공
  - ✓ LineNumberReader
- 바이트 스트림과 캐릭터 스트림의 호환
  - ✓ InputStreamReader, OutputStreamWriter

5) LineNumberReader 클래스

- 텍스트 파일을 라인 단위로 읽어 들이는 메소드 제공
- BufferedReader의 서브 클래스
- 주요 메소드
  - ✓ String readLine( )
  - ✓ int getLineNumber( )

```
File file = new File("src\\LineNumberTest.java");
FileReader fr = new FileReader(file);
LineNumberReader rd = new LineNumberReader(fr);
String line;
while ((line = rd.readLine( )) != null) {
	System.out.print(rd.getLineNumber( )+" ");
	System.out.println(line);
}
```

6) InputStreamReader 클래스

- 바이트 입력 스트림을 캐릭터 입력 스트림으로 바꾸기 위한 클래스
  - ✓ 바이트 단위로 읽은 후 문자로 바꾸어 처리함
- 생성자는 InputStreamReader(InputStream in)
- int read( )
  - ✓ 1개 문자를 읽어 리턴함
- int read(char[] cbuf, int offset, int length)

```
InputStreamReader isr = new
       InputStreamReader(System.in);
try {
   while((i = isr.read( )) != '끝') {
       System.out.print((char) i);
   }
} catch ( … …
```

Java프로그래밍

다음시간안내

10강. java.nio 패키지의 활용