

# ምክራፍ 11

## የጃቫ ገቢና ወጪ ምንጮች (Java IO Streams)

### 10.1 ምዕላደ - ቃላት

አማርኛ ቃል	እንግሊዘኛ ቃል
ሆሄ/ሆሄያት	character/characters
ተጠቃሚ	user
ገቢ	input
ወጪ	output
ምንጭ	stream
ኪቦርድ*	keyboard
ገጽ	screen
ፋይል*	file
የፋይል ጠቋሚ	file pointer
የዴታ ዓይነታት	data types
የመስመር ጫፍ	end of line
አዲስ መስመር	new line
የፋይል መጨረሻ	end of file
ኧክሴፕሽን*	exception

### 10.2 ምንጮች

በጃቫ እንዴት አድርገን ነው ዴታ ከኪቦርድ ፣ ከፋይሎች ፣ ወይም ከሌሎች ምንጮች የምናነበው? በተጨማሪ ወደ ፋይሎች ፣ ወደ ገጾች ፣ ወይም ወደ ሌሎች ምንጮች ዴታ እንዴት እንጽፋለን? ባለፉት ምዕራፎች የቀረቡት አንዳንድ ምሳሌዎች እነዚህን ጥያቄዎች በጨረፍታ ይነካሉ ፤ ነገር ግን በቂ መልስ አይሰጡም። በዚህ ክፍል መልሶቹን ለመልከት እንሞክራለን።

ምንጭ (**stream**) ዴታዎች ወይም ነገሮች የሚመጡበት ወይም የሚወጡበት አድረገን መገንዘብ እንችላለን። ተጠቃሚዎች ኪቦርድ ተጠቅመው ዴታዎችን ወይም ነገሮችን ካሰገቡ ፣ ኪቦርዱን የገቢ ምንጭ (**input stream**) ብለን እንጠራዋለን። እንደ ኪቦርድ ተመሳሳይ ተግባር ያላቸው የኮምፕዩተር መሣሪያዎችን በውሎ በዚህ ዘይቤ እንጠቅሳቸዋለን።

ባንጻሩ የወጪ ምንጭ (**output stream**) ብለን የምንጠራቸው የኮምፕዩተር አካሎችና መሣሪያዎች አሉ። ፋይሎች ፣ የሞኔተር ገጾች አብዮቹ ናቸው።

የምንጭ ቃል አጠቃቀሚዎችን ከተለመደው ትርጉሙ ላይ አዲስ ቃና ይጨምራል። ለፕሮግራሞች እነኪቦርድና የመሳሰሉት የገቢ ምንጭ ፤ እነገጽና የመሳሰሉት ደግሞ የወጪ ምንጭ ከተባሉ ፣ «ምንጭ» ሁለት አቅጣጫ ይኖረዋል። ለመጨረሻ ዴታዎችና ነገሮች መነሻ ፤ ለውጪዎቹ ደግሞ መድረሻ ለማለት ነው።

በጃቫ አሠራር ማንኛውም የገቢ ሆነ የወጪ ምንጭ በመደብ ርቢ መወከል አለበት። ከዛ በኋላ ነው ለሥራ ዝግጁ የሚሆነው። በተከታዩ ክፍል ዝርዝሩን እናያለን። በተጨማሪ የጠጉ ችግሮችን ይቀንስ ይረዳ ዘንድ የጃቫ የገቢም ሆነ የወጪ ምንጭ መደቦች

ግዴታዊ ሽኩሴ፣ ፕሮጀክቶች ይጠቀማሉ። በትክክል ተቀብሎ ፣ የተፈለገውን እርምጃ መውሰድ የውዴታ ግዴታ ነው።

### 10.3 የሆሜድት የገቢ ምንጮች

በምዕራፍ ፫ እንደተጠቀሰው ፣ የጃቫ ሆሜ የዩኒኮድን ደንብ ይከተላል። በመሆኑም እያንዳንዱ ሆሜ የሁለት ባይት ቦታ ይወስዳል። በሌላ በኩል የዩኒኮድ ሆሜ የማይጠቀም ማንኛውም ኦ/ሲ (Operating System) እንዲሁም ሶፍትዌር ከዚህ ጋር ስለማይጣጣም ከአንድ ስልት ወደሌላኛው በሁለቱም አቅጣጫ ቅየራ መካሄድ አለበት። በዚህ ክፍል ይኸን ሁኔታ ተረድተው ፣ ተገቢውን ቅየራ እየደረጉ የማንበብ ስራችንን የሚረዱ መደቦች እንጠቀማለን። ወደ ሌሎቹ የጃቫ የዴታ ዓይነቶች ወደ ፊት እንደርሰብታለን።

በመጀመሪያ ከኪቦርድ ሆሜድትን እንዴት እናንባለን? ከእንደዚህ ዓይነት የገቢ የኮምፕዩተር ክፍሎች ዴታ የምናነብበት ጥቂት ዘዴዎች አሉ። አንዱ ይኸን ይመስላል። በቅድሚያ ድርጊቶቹ በተርታ።

- አንድን ኪቦርድ በSystem.in ርቢ መወከል እንችላለን።
- ከኪቦርዱ ሆሜድትንና ሽሰትሪንግ ለማንበብ የInputStreamReader መደብ እንጠቀማለን
- እያንዳንዱን ፊደል ለማንበብ የዚህን መደብ read() ተብሎ የሚጠራ መላ እንጠቀማለን።

እነዚህ ድርጊቶች በተግባር ያካተተው ከዚህ በታች ይታያል። አንባቢው የደመቁትን ሐረጎች ከተመለከተ ፣ ሦስት ነገሮችን ይታዘባል። የመጀመሪያው ፣ ስህተት ተፈጥሮ ሽኩሴ፣ ፕሮጀክቶች ከተወረወረ ፣ የሜይን መላ እሱን ለጠራው ወገን ያንን ሽኩሴ፣ ፕሮጀክቶች ያስተላልፋል። ይህ አንድ ዓይነት እርምጃ መሆኑን ለማሳየት እንጂ ተግባሩ የግዴላሽነት ነው። ምንም ችግር ይምጣ ፣ የራሴን ጣባ ሌላው ወገን አስታቅፋለሁ እንደማለት ነው። አንዳንድ ሁኔታዎች ውስጥ ግን መልካም እርምጃ ሊሆን ይችላል። ቀጥሎ ፣ በSystem.in የተመሠረተ ከኪቦርድ አንባቢ ርቢ ይፈጥራል። በመጨረሻ ከኪቦርድ መጨዎቹ ሆሜድትን አንድ በአንድ ያንባል።

```

ከኪቦርድ ሆሜድት አንባቢ መደብ

import java.io.* ;
public class KeyboardReader {

    /** Method: An entry point for program execution */
    public static void main(String[] args) throws IOException {
        InputStreamReader ir = new InputStreamReader(System.in) ;
        String data = "" ;

        // read until no more character
        int c = ir.read() ;
        while (c != 13) {
            data += (char) c ;
            c = ir.read() ;
        }
        System.out.println("Entered data was: " + data) ;
    }
}

Download: KeyboardReader.java

```

ይህ ፕሮግራም ከኪቦርድ ሆሜድትን ማንበብ መቻሉ ባያጠራጥርም ፣ ፍጥነቱን ማሻሻል እንዲሁም ሽሰትሪንግ ደጋግሞ ከመፍጠር መቆጠቡ ጥሩ ስራ ነው። ሆሜድቱን አንዳንድ በተርታ ከማንበብ ይልቅ በሽሰትሪንግ መልክ አንድ ጊዜ ማንበቡ ይመረጣል። ይኸንን የምናደርገው የBufferedReader መደብ በመደረብ ነው። ቃል አጻጻፉ እነሆ። አደራረቡን በጥሞና እንመልከት።

```

InputStreamReader ir =
    new InputStreamReader(System.in) ;
BufferedReader br = new BufferedReader(ir) ;

```

አሁን የreadLine() መላ ከbr ርቢ ፣ በአንድ ተናጠል አርምጃ ከኪቦርድ የሚመጡትን ሆሂያት በመሉ ያነባል።

```
String line = br.readLine() ;
```

ቀጥሎ እነዚህን ለውጦችና የተሻሻለ የአክሲዮን አያያዝን ያከተሉን ፕሮግራም አንይ።

**ከኪቦርድ ሆሂያት አንባቢ መደብ**

```

import java.io.* ;
public class KeyboardReaderBetter {

    /** Method: An entry point for program execution */
    public static void main(String[] args) {
        InputStreamReader ir = new InputStreamReader(System.in) ;
        BufferedReader br = new BufferedReader(ir) ;
        String data = "" ;

        // read until no more character
        try {
            data = br.readLine() ;
        } catch (IOException e) {
            System.out.println(e) ;
        }
        System.out.println("Entered data was: " + data) ;
    }
}

```

[Download: KeyboardReaderBetter.java](#)

አሁን ወደ ፋይል የገቢ ምንጮች እናመራለን። የማንበብ ሂደቱ ቅደም-ተከተል ይኸን ይመስላል።

- ፋይል ከፍቶ ፣ አንብቦ ፣ በመጨረሻ ለመዝጋት በFile ርቢ የግድ መወከል አለበት።
- ሆሂያትን አንድ በአንድ ለማንበብ የFileReader መደብ እንጠቀማለን
- ነገር ግን የንባቡን ሥራ ለማጣደፍ የBufferedReader መደብን ከላይ እንደርብና መስመር በመስመር የፋይሉን ያዘት እናንባለን።

**ከፋይል ሆሂ አንባቢ መደብ**

```

import java.io.* ;
public class FileCharacterReader {

    /** Method: An entry point for program execution */
    public static void main(String[] args) throws IOException {
        // instantiate String objects
        String fileName = "text.txt" ;
        String cities = "" ;
    }
}

```

```

// instantiate and prepare character stream objects
File inFile = new File(fileName) ;
FileReader fr = new FileReader(inFile) ;
BufferedReader br = new BufferedReader(fr) ;

// read each line from the file
String line = br.readLine() ;
while (line != null) {
    cities += line + '\n' ;
    line = br.readLine() ;
}
System.out.println("Contents of " + fileName) ;
System.out.println(cities) ;
}
}

```

[Download: FileCharacterReader.java](#)  
[Download: text.txt](#)

እንደተለመደው ደምቀው በሚታዩት መስመሮች ላይ እናተኩር። የመጀመሪያዎቹ ቃሎች ሦስት ልዩ ልዩ ርቢዎች ከሚከተሉት መደቦች ይፈጥራሉ፡ የFile መደብ ከፋይሎች ጋር አብሮ ለመሥራት (2ኛ) የFileReader መደብ መሠረታዊውን ንባብ ለማካሄድ (3ኛ) የBufferedReader መደብ ንባቡን ለማቀጠፍ። ሦስቱን ቃሎች አሳጥሮ እንደዚህ መጻፍ ይቻል ነበር ፤ ግን የአንባቢውን ምኞት ይቀንሳል።

**አጭር አጻጻፍ**

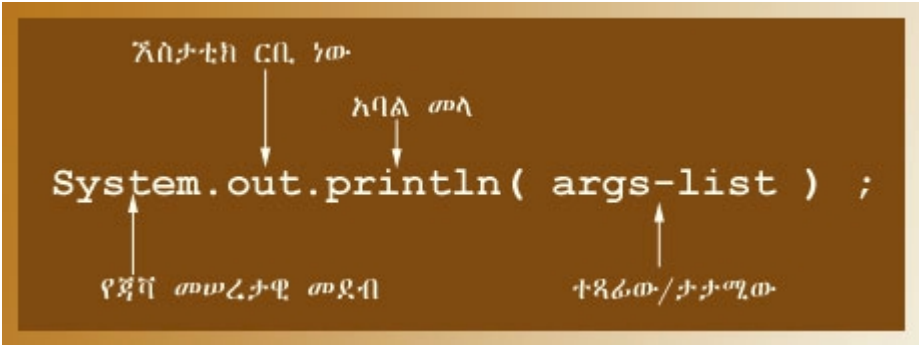
```

BufferedReader br = new BufferedReader(
    new FileReader(
        new File(fileName) ) ) ;

```

### 10.4 የሆሄያት የወጪ ምንጮች

አሁን ትኩረታችንን ወደ የሆሄያት ወጪ ምንጮች እናዘራለን። ነገሮችን ወደ ገጽ መጻፍ ቀላልና ጥናቱ ከተጀመረበት ጊዜ እስከሁን ድረስ ስናካሄደው ቆይተናል። ስለዚህ ወደ ፋይሎች ሆሄያትን እንዴት እንጽፋለን የሚለውን ጥያቄ ለማየት እንሞክራለን። ወደ እሱ ከመግባታችን በፊት ግን የሚከተለውን ቃል ምን እንደነበረ በዝርዝር እንመርምር።



በተሰጡት ምሳሌዎች በሙሉ ይህ ሳይጠቀስ አላለፈም። የSystem መሠረታዊ ከሚባሉት የጃቫ መደቦች አንዱ ነው። ከኪርርድ ሆኔያትን ለማንበብ ፣ ወደ ገጽ ነገሮችን ለመጻፍ ፣ ስዓት ለማወቅ ፣ እንዲሁም ለሌሎች ተግባሮች እንጠቀመዋለን። ይህ መደብ በይፋ ያወጣቸው የኧስታቲክ ተውላጠ-ቃላት አሉት፦ **err** ፣ **in** እና **out** ናቸው። የ**out** ርቢ የ**PrintStream** መደብ ዘርና የተለያዩ የደታ ዓይነቶችን ወደ ገጽ ለመጻፍ የሚያስችሉ መላዎች ያሉት ነው። አሁን ወደ ፋይሎች ጉዳይ እንሂድ።

ከፋይሎች ማንበብ ወይም ወደ እነሱ መጻፍ ፣ የFile ርቢ ውክልና ማስፈለጉን አስቀድመን አውስተናል። ስለዚህ የመጀመሪያው ተግባራችን ይኸው ይሆናል።

```
File outFile = new File("planets.txt") ;
```

ይህ ፋይል ከሌለ አዲስ ይፈጠራል ፤ ካለ ግን ይዘቱ ይሰረዘና በአዲሱ ይተካል። በነገራችን ላይ ፣ የFile መደብ በራሱ የማንበብ ሆነ የመጻፍ ችሎታ ስለሌለው ጸሐፊ መደብ መደረብ ይኖርብናል ፤ እናም እነሆ።

```
Filewriter fr = new FileWriter(outFile) ;
```

የFileWriter መደብ ሆኔያትን የመጻፍ ችሎታ አለው ፤ ግን ስራው ይጣደፍ ዘንድ የBufferedWriter መደብ መደረብ ይረዳል።

```
BufferedWriter bw = new BufferedWriter(fr) ;
```

እነዚህን ሁሉ አካቶ ወደ ፋይል የመጻፍን ስራ የሚያካሂደው መደብ ይህ ነው።

ሆኔያት ጸሐፊ መደብ
<pre>import java.io.* ; public class FileCharacterWriter {      /** Method: An entry point for program execution */     public static void main(String[] args) throws IOException {         // instantiate array of String objects         String[] <b>planets</b> = {"Mercury", "Venus", "Earth",                              "Mars", "Jupiter", "Saturn",                              "Uranus", "Neptun", "Pluto" } ;          // instantiate and prepare character stream objects         <b>File outFile = new File("planets.txt") ;</b>         <b>FileWriter fw = new FileWriter(outFile) ;</b>         <b>BufferedWriter bw = new BufferedWriter(fw) ;</b>          // write the array contents into an output file         try {             for (int i=0; i &lt; planets.length; i++) {                 <b>bw.write(planets[i]) ; // write string</b>                 <b>bw.newLine() ; // write newline</b>             }             bw.close() ; // close file         } catch (IOException e) {             System.out.println("File write failur") ;         }     } }</pre>

[Download: FileCharacterWriter.java](#)

## 10.5 የጃቫ ዴታ ዓይነታትና ምንጮች

በሌሎች የኮምፕዩተር ቋንቋዎች ልዩ ልዩ የዴታ ዓይነታትን ፣ ማለት char ፣ int ፣ float ፣ double እና የመሳሰሉትን በቀላሉ መጻፍ ወይም ማንበብ ይቻላል። ነገር ግን ወደ ጃቫ ስንመጣ ሁኔታው ለየት ይላል። አንዳንድ የጃቫ ዴታ ዓይነታት ከሌሎች ቋንቋዎች ጋር ሲነጻጸሩ ልዩነት እናገኛለን። ለምሳሌ ያህል ፣ የድሮ ቋንቋዎችና ኦ/ሲኾ የዩኒኮድን ሆሂዶት አይደግፉም ፤ ማለትም አሁንም ለአንድ ሆሂ የሚሰይሙት ቦታ አንድ ባይት (Byte) ነው። ለምሳሌ ዊንዶስ 98 ብንወስድ ይህ ገሀድ ይሆናል።

C++ መሠረታዊ የዴታ ዓይነታት አውንታዊ ዕቤት ብቻ ወይም አውንታዊና አሉታዊ ዕቤቶችን እንዲይዙ ይፈቅዳል። በጃቫ አውንታዊ ብቻ የሚለው አማራጭ የለም። hchar በስተቀር ሁሉም አውንታዊና አሉታዊ ዕቤቶችን የመጠበቅ ኃላፊነት አለባቸው።

ይኸን ችግር እዚህ ማንሳቱ ዋጋው ምንድን ነው? በC++ የተጻፈ ፕሮግራም ያወጣውን ፋይል የተለያዩ የጃቫ ዴታ ዓይነታትን ተጠቅመን እናንብብ ብንል አዘቅት ውስጥ እንገባለን። ዴታዎች ይዘበራረቃሉ። ማንነታቸው ጠፍቶ ትርጉም ያጣሉ። በእነዚህና በመሳሰሉት ምክንያት ልዩ ልዩ የጃቫ ዴታ ዓይነታትን ማንበብ ወይም መጻፍ ከፍተኛ ጥንቃቄ ይጠይቃል። ሊነሱ የሚችሉ ችግሮችን ሙሉ በሙሉ ለመቀነስ ፣ መፍትሔው ከሆሂዶት ውጭ ንባብ ጠያቂ ፋይሎችን በጃቫ ካልተጻፉ በስተቀር አለመሞከር።

ይኸንኑን ምክር በመከተል ፣ መጀመሪያ በምዕራፍ ፫ የተሰጠውን የጨረቃ ዴታ ወደ ፋይል የሚጽፍ መደብ እናቀርብና አስከትሎ ያንን ፋይል መልሶ እንባቢ መደብ እናያለን።

እንደ ተለመደው ወደ ፋይል ለመጻፍ የFile ርቢ እንፈጥራለን። ቀጥሎ ከፋይል ጋር መሥራት ያለበትን መደብ ርቢ እናበጃለን። በመጨረሻ ፣ የጃቫ ዴታ ዓይነታት መጻፍ የሚችለውን መደብ በመደረግ ዝግጅቱን እናጠናቅቃለን።

```
File in = new File("moon.txt") ;
FileOutputStream fos = new FileOutputStream(in) ;
DataOutputStream dos = new DataOutputStream(fos) ;
```

የDataOutputStream መደብ ማንኛውንም መሠረታዊ የጃቫ የዴታ ዓይነት መጻፍ ይችላል። ለእያንዳንዳቸው የራሱን ጸሐፊ መላ ይለግሳል። ለምሳሌ የdouble ዴታ መጻፍ ብንሻ ውጤቱ ይኸን ይመስላል።

```
final double pi = 3.14159 ;
dos.writeDouble(pi) ;
```

ተግባራዊ ምሳሌ፦

```
የጃቫ ዴታ ዓይነታት ጸሐፊ

import java.io.* ;
public class DataWriter {

    /** Method: An entry point for program execution */
    public static void main(String[] args) throws IOException {
        double[] data = {7.36E22, 1738, 3340, 1.67, 3.80E5} ;

        // instantiate and prepare character stream objects
        File outFile = new File("moon.txt") ;
```

```

FileOutputStream fos = new FileOutputStream(outFile) ;
DataOutputStream dos = new DataOutputStream(fos) ;

// write the array contents into an output file
try {
    for (int i=0; i < data.length; i++) {
        dos.writeDouble(data[i]) ;
    }
    dos.close() ;           // close file
} catch (IOException e) {
    System.out.println("File write failur") ;
}
}
}

```

[Download: DataWriter.java](#)

መደቡ በእራህ ውስጥ የተጠበቁትን የጨረቃ ዴታዎች ወደ «moon.txt» ይጽፋል። የተጠቀሙናቸውን መደቦችና ስማቸውን አንባቢው በጥንቃቄ ሊመረምራቸው ይገባል። የተጻፈውን ፋይል ቀጥሎ የቀረበው መደብ አንብሎ ውጤቱን ገጽ ላይ ያሳያል።

#### የጃቫ ዴታ አንባቢ መደብ

```

import java.io.* ;
public class DataReader {

    /** Method: An entry point for program execution */
    public static void main(String[] args) throws IOException {

        // instantiate and prepare byte stream objects
        File inFile = new File("moon.txt") ;
        FileInputStream fis = new FileInputStream(inFile) ;
        DataInputStream dis = new DataInputStream(fis) ;

        // Read from the file and print the content
        boolean next = true ;
        try {
            while (next) {
                System.out.println(dis.readDouble()) ;
            }
            dis.close() ;           // close file
        } catch (IOException e) {
            // do nothing
        }
    }
}

```

[Download: DataReader.java](#)

To contact: [info@senamirmir.com](mailto:info@senamirmir.com)

Copyright © 2002-2005 Senamirmir Project