ምእራፍ ፫ ዴታ ዓይነታት

የኮምፕዩተር ሥራዎች ሲከናወኑ ተካፋይ ፕሮግራሞች ፥ ዴታዎችንና የመሳሰሉትን የሚጠበቁበት ቦታ የማስታወሻ ክፍል (memory) ይባላል። ፕሮግራሞች ለሥራ እንዲሰማሩ ስንጠይቅ ወደ ማስታወሻ ክፍል ውስጥ እንዲገቡ ይደረግና ለኮምፕዩተሩ ሒሊና ቀርበው ሥራዎቻቸውን ያከናውናሉ። ገቢና ወጪ ዴታዎችን በሥራ ወቅት የማስታወሻ ክፍል ውስጥ ይጠበቃሉ። ዴታዎች አይነትና ጠባይ አላቸው። ፕሮግራሞች ባያያዝና አጠቃቀም ያንን ጠንቅቀው መረዳትና ማክበር አለባቸው። በዚህ ክፍል ውስጥ ለማየት የምንሞክረው ይኸንኑ ነው።

3.1 ዴታ ምንድን ነው?

በመሠረቱ ዴታ ሲባል ያልተበረዘ ፥ ያልተዘነቀ ፥ ወይም ያልተቀየጠ የሚነገር ወይም የሚጻፍ እርግጠኛ መረጃ። ስለድርጊት ፥ ስለቦታ ፥ ስለሂደት ፥ ስለሕዝብ ፥ ስለሁኔታና የመሳሰሉትን ይጨምራል። ስም ፥ ትውልድ ቦታ ፥ ዕድሜ ፥ ሥራ እንዲሁም ደመወዝ ፥ ስለአንድ ሰው ሁኔታ የሚነግሩን መረጃዎች ናቸው። የሚከተለው ወንጠረዥ ስለጨረቃ የተወሰነ ዴታ ይሠጠናል። እንደዚህ አይነቱ ዴታ ከማብራሪያ ወይም ገለጻ ጋር በተለያየ መልክ ሲቀርብ ዜና ፥ ዘገባ ፥ ወይም ኢንፎርሜሽን (information) ብለን እንጠራዋለን። ዴታና ኢንፎርሜሽን ሁለት የተለያዩ ነገሮች ናቸው።

ዴታ ስለጨረቃ	
የአካል ክብደት	7.36E22 kg
ሬድየስ	1738 km
አማካይ ደንስቲ	3340 kg/m^3
ስበት (gravity)	1.67 m/s^2
አማካይ ርቀት ከመሬት	3.80E5 km
Source: http://www.nineplanets.org	

ዴታዎች አይነት አላቸው። ተግባራቸውና የሚይዙት ቦታ አይነታቸውን ይወስናል። ባጠቃላይ ረገድ ፥ አፈራረጃቸው እንደሚከተለው ነው።

1. የቁጥር ዴታ አይነታት
2. የንባብ ዴታ አይነት
3. የቡሊኤን ዴታ አይነት
4. የመደብ ዴታ አይነታት ናቸው።

የጃቫ ዴታ አይነታት በሁለት ይከፈላሉ።

1. መሠረታዊ የዴታ አይነታት

አይነታት	መጠበቅ የሚችሉት
byte	ከ -128 እስከ 127

short	ከ -32768 እስከ 32767
int	ከ -2147483648 እስከ 2147483647
long	ከ -9223372036854775808 እስከ 9223372036854775807
char	ከ 0 to 65535 ወይም ከ '\u0000' እስከ '\uffff'
float	ከ -3.4E38 እስከ 3.4E38
double	ከ -1.7E308 እስከ 1.7E308
boolean	ከ true ወይም false አንዱ

2. መደባዊ የዴታ አይነታት (መደብ ጠባቂ)

3.2 መሠረታዊ ዴታ አይነታት

መሠረታዊ ዴታ አይነታት በጃቫ ቋንቋ የተደነገጉ ከመሆናቸውም በላይ የራሳቸው የሆነ ልዩ መጠሪያ ስም አላቸው። ተግባራቸውና አሠራራቸው ግልጽና በፍጹም የማያሻማ ነው። መለወጥ ወይም በሽራፊ መቀየር በጭራሽ ክልክል ነው ፤ ስማቸውን ለተለየ ነገር እስከመጠቀም ድረስ። ቀጥለን የመሠረታዊ ዴታ አይነታትን አጠርና ዘርዘር በሆነ አቀራረብ እንመለከታለን።

- ድፍን የቁጥር ቤተሰቦች **(integer)** የማይሸራረፉ ወይም ዴስማል ነጥብ የሌላቸው ናቸው። ለምሳሌ 128 ፣ 256 ፣ 512 ፣ 1024 እና የመሳሰሉት ድፍን ቁጥር ናቸው። እንደ 3.14159 ወይም 2.71 አይነቶቹ ግን ድፍን ቁጥር አይደሉም። በድፍን ቁጥር ሥር ያሉት አምስቱ የጃቫ የዴታ አይነታት እነዚህ ናቸው።

  byte, char, short, int, long

  ቁጥሮች ስለሚደመሩ ፥ ስለሚቀነሱ ፥ ስለሚባዙ ፥ ስለሚካፈሉ እንዲሁም በሌሎች ሥልያዎች ስለሚገቡ ይህ ጠባይ ልዩ ያደርጋቸዋል።

  የchar ዴታ የዩንኮድ ሆሄ እየተባለ ይጠራል። ምንም እንኳን የድፍን ቁጥር ቤተሰብ አባል ቢሆንም አብይ ተግባሩ ሆሄያትን መወከል ብቻ ነው። በዩንኮድ ደንብ መሠረት ፊደላት ከ0 በላይ እንጂ ፥ በታች ኮድ አይሰየሙም። በመሆኑም አሉታዊ ዕሴት ለchar መሰየም በፍጹም የተከለከለ ነው።

የኢትየጵያ ፊደል ዩንኮድ ሠንጠረዥ፦ UnicodeChar.java

```
public class UnicodeChar {

   /** An entry point for program execution */
   public static void main(String[] args) {
      // declares a character initialized to Ethiopic HA
      char fidel = '\u1200';

      // print Ethiopic Unicode(Version 3.0)
      while( fidel <= '\u137F' ) {
         for (int i=0; i < 8; i++) {
            System.out.print( (int) fidel + ", ");
            fidel++ ;
         }
         System.out.println();
      }
   }
}
```

Download: UnicodeChar.java

- ተግባራዊ የቁጥር ቤተሰቦች (floating point) ድፍን ቁጥር ብቻ ሳይሆን የተሸረፈም መያዝ ይችላሉ።

በቁጥርነታቸው ከድፍን ቁጥር ጋር አንድ አይነት ጠባይ ሲኖራቸው በሚይዙት የዕሴት አይነት ግን ይለያሉ። ጃቫ ሁለት የተግባራዊ የቁጥር ዴታ አይነት አሉት።

```
float, double
```

- ቡሊኤን (boolean) ሁለት አቋሞች አሉት፦ እውነትና ሐሰት። እነዚህ ሁኔታዎች በተለያየ አባባል ሲገለጹ ይሰማል። ለምሳሌ «አለ ፥ የለም» ፤ «አንድ ፥ ዚሮ» ወይም «በራ ፥ ጠፋ»።

```
boolean
```

3.3 መደባዊ ዴታ

መደባዊ ዴታ ማለት የመደብ ዘር ነው። እያንዳንዱ መደብ የሩሱ የሆነ ማንነት አለው። ራሱን ከሌሎች መደባት ጋር ማዘመድ ይቻላል። ካደረገ ፥ የተዛመደውን መደብ ማንነት በራሱ ላይ ይጨምራል።

መደብ ከፈጠርን ቡኋላ ወደሥራ ላይ ለማዋል ማድረግ ካሉብን ነገሮች መካከል አንዱ የመደብ ርቢ (objects) መፍጠር ነው። «የመደብ ርቢ» ወይም object አንድ መደብ በሥራ ሲውል የሚከሰት ነገር ነው።

መደባት በውስጣቸው «እስታቲክ» ተውላጠ ቃላት ወይም ተግባራት (static variables or functions) ካልያዙ ፥ እነሱን ለመጠቀም የግድ የመደብ ርቢ (object) መፈጠር አለበት። ለምሳሌ የጃቫ እስትሪንግ መደብ ከንባብ ጋር የተያያዙ ችግሮችን መፍትሔ ለመስጠት የምንጠቀመው መደብ ነው። በሥራ ላይ ለማዋል ግን የግድ ርቢ ማውጣት አለብን። በእርግጥ ያለመደባዊ ርቢ የእስትሪንግን መደብ መገልገል የምንችልበት ሁኔታ አለ። ነገር ግን በጣም ውስን ነው።

```
String city = new String("Addis Ababa") ;
```

በዚህ ቃል መሠረት፦ «city» የእስትሪንግ የመደብ ርቢ ነው። ጠባያቱና ተግባራቱ እንደ መደቡ ይሆናል። ይዘቱ «Addis Ababa» ነው። የመደብ ርቢ ያለ ይዘት መፍጠር ይቻላል። ለምሳሌ፦

```
String continent ;
```

ይህ የመደብ ርቢ ፥ ለጊዜው ይዘት ባይኖረውም እንደተፈለገው አዲስ ይዘት መሰየም ይቻላል። የሚከተለው ቃል ይኸን ያሳያል።

```
continent = new String("Africa") ;
```

የመደብ ርቢ የዴታ ማንነታቸው በመደባቸው ይወሰናል። ጠባያቸውና ተግባራቸው የመደባቸው ነጸብራቅ ነው።

3.4 ተውለጠ-ቃላት (variables)

ለማንኛውም የኳድራቲክ ጥያቄ አጠቃላይ መፍትሔ ማግኘው ፎርሙላ የሚከተለው ነው።

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

ይህ ፎርሙላ ለማንኛውም ተግባራዊ ቁጥር ያገለግላል። ምክንያቱም ሦስቱ ምልክቶች «a» ፥ «b» ፥ እና «c» ልዩ ልዩ ዕሴቶችን

ማቀፍ ስለሚችሉ። እንደዚህ አይነት ምልክቶች በኢንግሊዘኛ «**variables**» ተብለው ይጠራሉ። እዚህ «ተውላጠ-ቃላት» ብለን እንጠራቸዋለን።

ይኸን ፎርሙላ በሥራ ላይ ያዋለ የጃቫ ፕሮግራም እንጻፍ ብንል ፥ የግድ ሦስቱን ዕሴቶች ተቀብለን የምንጠብቅባቸውና ስሌቱን የምንፈጽምበት ተውለጠ-ቃላት ያስፈልጉናል። በተጨማሪ ፎርሙላው የሚሰጠውን ሁለት ውጤቶች የምናስቀምትበት ሌሎች ተውላጠ-ቃላት ማከል አለብን።

መሠረታዊ ዴታ አይነታትና ተግባራቸው፦ Operations.java

```
import java.util.* ;

public class Operations {

   /** An entry point for program execution */
   public static void main(String[] args) {
      int left, right, state ;
      left = right = state = 0;   // allowed in Java

      Random rand = new Random() ;
      left  = rand.nextInt(15)  ;
      right = rand.nextInt(15) ;

      System.out.println("Java Bitwise operation");
      state = left & right;
      System.out.println(left + " & " + right + " = " + state);

      state = left | right;
      System.out.println(left + " | " + right + " = " + state);

      state = left ^ right;
      System.out.println(left + " ^ " + right + " = " + state);
   }
}
```

Download: Operations.java

ይህ ምን ማለት ነው? ገቢና ወጭ እንዲሁም ውስጣዊ ዴታዎች የማስታወሻ ክፍል (memory) ይጠይቃሉ። እነሱን ለመጠበቅ ፥ አልፎ ለመገልገል ተውላጠ-ቃላት (variables) መውጣት አለባቸው። ከዚህ በመነሳት ተውላጠ-ቃላት ከዴታ አይነታትና ከማስታወሻ ክፍል ጋራ ያላቸው ግንኙነት አንዱ ባንዱ ላይ ተመክቶ መሥራት ብቻ ሳይሆን አንደኛው ሌላኛውን እኩል በሆነ መንገድ ይገልጻሉ። ለምሳሌ፦ የኢንትጀር ተውላጠ-ቃል ስንል ፥ የኢንትጅር ዴታ አይነት ለማለት ነው። በተጨማሪ የኢንትጀር ዴታ አይነት ልኩ አራት ባይት (byte) ነው ስንል ፥ ከማስታሻ ክፍል የሚይዘው ቦታ አራት ባይት ለማለት ነው። የተውላጠ-ቃላት አጠቃላይ ባህሪ እንዚህ ናቸው።

1. ተውላጠ-ቃላት በሚጠብቁት የዴታ አይነት ይፈጠራሉ። ምንና ማንነታቸው እንዲሁም ተግባራቸው በዴታው አይነት ይደነገጋል።
2. ተውላጠ-ቃላት ሲወጡ ፥ ዴታ መጠበቂያ ቦታ ከማስታወሻ ክፍሉ (memory) ይከፈታላቸዋል ፤ ነገር ግን ይሕ ለመደባዊ ተውላጠ-ቃላት አንድ አማራጭ ነው። የተከፈተው የማስታወሻ ክፍል ተጠቃሚና ተገልጋዮቹ እነሱ ተውላጠ-ቃላቱ ናቸው።
3. ተውላጠ-ቃላትን በወጡበት የዴታ አይነት መሠረት መጠቀም ይፈቀዳል። ለምሳሌ፦ በኢንትጀር ተውላጠ-ቃል ውስጥ ድፍን ቁጥሮችን መጻፍ ወይም ማንበብ ፥ መሠረዝና ልዩ ልዩ የሂሳብ ተግባራትን መፈጸም ይቻላል።

4. የተውላጠ-ቃላትን ይዘት ከአንድ የታላቅ የዴታ አይነት ወደ ታናሽ ወይም በዝውሩ መቀየር የሚፈቀድበት ሁኔታ አለ።

3.4.1 የመሠረታዊ ዴታ ተውላጠ-ቃላት አወጣጥ

- የ**char** ተውላጠ-ቃል

```
char fidel ;
```

«**char**» የዴታ አይነት ሲሆን «**fidel**» ደግሞ ተውላጠ-ቃል ነው። አንድ ተውላጠ-ቃል ሲወጣ መነሻ ዕሴት መሰየም ይቻላል። ለምሳሌ፦

```
char fidel = 'ሀ';
```

የ«**fidel**» ተውላጠ-ቃል መሸከም የሚችለው አንድ ሆሄ ብቻ ነው። የያዘውን ሆሄ መቀየር የምንሻ ከሆነ ተግባሩ ቀላል ነው።

ከዚህ ቃል በኋላ የ«fidel» ዕሴት «ለ» ይሆናል። እንደጋና ሌላ ብንሞክር

```
fidel = 'A' ;
```

መሠረታዊ ዴታ አይነታትና ተግባራቸው፦ CharReader.java

```
public class CharReader {

   /** An entry point for program execution */
   public static void main(String[] args) throws Exception {
      System.out.println("Enter characters") ;  // prompt user
      int c1 = System.in.read() ;               // read 1st char
      int c2 = System.in.read() ;               // read ahead
      while ( true ) {                          // loop all  time
         if (c2 == 13) {                        // if \n, end loop
            System.out.print( (char) c1) ;      //
            break ;                             // terminate loop
         }
         System.out.print( (char) c1 + ", ") ;  //
         c1 = c2 ;                              // for read ahead
         c2 = System.in.read() ;                // read ahead
      }
   }
}
```

Download: CharReader.java

- የ**int** ተውላጠ-ቃል

```
int x; int y; int z;
```

ሦስት ተውላጠ-ቃላት ይፈጠራሉ። ባንድ መስመር ወይም በተለያየ መስመር ላይ መጻፋቸው የሚያመጣው ለውጥ የለም። ሦስቱም ተውላጠ-ቃላት የዴታ አይነታቸው ተመሳሳይ ስለሆነ ፥ ባንድ ወገን ማውጣት እንችላለን።

```
int x, y, z ;
```

ይህ ከላይ ካየነው በውጤት ደረጃ ልዩነት የለውም ፤ ባቀማመጥ እንጂ። ፕሮግራም ጸሐፊዎች ተውላጠ-ቃል ሲያወጡ መነሻ ዕሴት መሰይም ይመርጣሉ። ተግባሩን በእንግሊዘኛ initialization ብለው ይጠሩታል።

```
int x = 0, y = 0, z = 0 ;
```

መነሻ ዕሴቱ ዜሮ ከሆነ በጃቫ ረገድ ምንም ፋይዳ የለውም። ምክንያቱም ጃቫ ተውላጠ-ቃላት ለሥራ መከሰማራቸው በፊት ራሱ መነሻ ዕሴት ሰለሚሰይማቸው።

- የ**double** ተውላጠ-ቃል

```
double ቁጥር = 2.5 ;
```

የዴታው አይነት «**double**» ፥ የተውላጠ-ቃሉ ስም «ቁጥር» እንዲሁም መነሻ ዕሴቱ 2.5። የdouble ተውላጠ-ቃል መጠን 8 ባይቶች ነው። ድፍን ወይም ደሲማል ቁጥሮችን መሸከም ይችላል።

3.4.2 የመደባዊ ተውላጠ-ቃላት አወጣጥ

መደባዊ ተውላጠ-ቃላት የባለቤታቸው «መደባዊ-አካል» ጠባቂዎች ናቸው። ይኸን ጉዳይ በቅርብ ለማየት ይረዳን ዘንድ የFibonacciSeqeunce መደብን እንጠቀም። ሙሉ የመደቡ ኮድ ታች ይገኛል። ይኸን መደብ ለመጠቀምና በሥራ ላይ ለማዋል የግድ መደቡን ማርባት አለብን። ተፈጣሪው ፥ መደባዊ ተውላጠ-ቃል ይሆናል። የመደብ ርቢ ወይም መደባዊ ተውላጠ-ቃል ስንል የመደብ ርባታዎች ለማለት ነው። የሁለቱ ስሞች ተግባራዊ ትርጒም ተመሳሳይ ነው።

```
FibonacciSequence fibonacci = new FibonacciSequence();
```

ከዚህ ውስጥ «**fibonacci**» የሚለው ተውላጠ-ቃል የFibonacciSequence ርቢ ነው ፤ መደቡ ተግባራዊነቱን ለመግለጽ በቅድሚያ ከሚጠይቃቸው ሂደቶች መካካል አንዱ። ይኸንን ቃል በመንተራስ የ**sequence()**ን መላ በመጥራት የፊቢናቺን ተከታታይ ቁጥሮች ማውጣት እንችላለን።

```
fibonacci.sequence(10);
```

በዚህ ቃል መሠረት የተጠራው ተግባር ዐሥሩን የመጀመሪያ የፊቢናቺን ቁጥሮች ያወጣል። ይኽ መሠረታዊ የመደባዊ ተውላጠ-ቃል አቋቋምና አሠራር ነው። ለማከል ያህል ፥ መደባዊ ተውላጠ-ቃል ያለይዘት መፍጠር ይፈቀዳል። በጃቫ ሕግ አይከለከለም። ያለይዘት ፈጥሮ እንዳሥፈላጊነቱ ወደፊት ይዘቱን በመጨመር መጠቀም የተለመደ አሠራር ነው። አወጣጡ ይኸን ይመስላል።

```
FibonacciSequence fibonacci;
```

አስከትሎ ይዘቱን በዚህ መንገድ መፍጠር ይቻላል።

```
fibonacci = new FibonacciSequence();
```

ይህ አሠራር ዘላቂ ጥቅም አለው። አንድን መደባዊ ተውላጠ-ቃል ፈጥሮ እንዳሥፈልጊነቱ በየጊዜው ይዘቱን መቀያየር ይቻላል። ሌላኛው ፥ ዝምድና ያላቸውን መደባት ለማገናኘትና ለመጠቀም ይረዳል። ይኸን የመጨረሻውን ወደፊት እንደርስበታለን።

አንባቢው በጥብቅ መጠንቀቅ ያለበት ነገር ቢኖር መደባዊ ተውላጠ-ቃላትን ያለይዘት ከመጠቀም ነው። ይዘት የሌለው መደባዊ ተውላጠ-ቃል ይዞ መላዎችን (functions) መጥራት ወይም አባላቶቹን ለማየት መሞከር ከፋኛ ስህተት ነው።

የፊቢናቺ ቁጥሮች፦ FibonacciSequence.java

```
public class FibonacciSequence {

   /* Generates as many as n the first fibonacci sequence */
   void sequence(long n) {
      long f1 = 0, f2 = 1, f3 = 0 ;

      System.out.print("Fib:  " + f1 + ", ");
      for (int i=0; i < n; i++) {
         f1 = f2 ;
         f2 = f3 ;
         f3 = f2 + f1 ;
         System.out.print(f3 + ", ");
      }
   }

   /** An entry point for program execution */
   public static void main(String[] args) {
      FibonacciSequence  fibonacci = new FibonacciSequence();
      fibonacci.sequence(10);
   }
}
```

Download: FibonacciSequence.java

To contact: info@senamirmir.com
Copyright © 2002-2005 Senamirmir Project