

ይታ ዓይነታት ምክራፍ

የኮምፕዩተር ሥራዎች ሲከናወኑ ተከፋይ ፕሮግራሞች ፣ ይታዎችንና የመሳሰሉትን የሚጠበቁበት ቦታ የማስታወሻ ክፍል (memory) ይባላል። ፕሮግራሞች ለሥራ እንዲሰማሩ ስንጠይቅ ወደ ማስታወሻ ክፍል ውስጥ እንዲገቡ ይደረግና ለኮምፕዩተሩ ሒሳብና ቀርቦው ሥራዎቻቸውን ያከናውናሉ። ገቢና ወጪ ይታዎችን በሥራ ወቅት የማስታወሻ ክፍል ውስጥ ይጠበቃሉ። ይታዎች አይነትና ጠባይ አላቸው። ፕሮግራሞች ባያያዝና አጠቃቀም ያንን ጠንቅቀው መረዳትና ማክበር አለባቸው። በዚህ ክፍል ውስጥ ለማየት የምንሞክረው ይኸንኑ ነው።

3.1 ይታ ምንድን ነው?

በመሠረቱ ይታ 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 ሠንጠረዥ ስለጨረቃ የተወሰነ ይታ ይሠጠናል። እንደዚህ አይነት ይታ ከማብራሪያ ወይም ገለጻ ጋር በተለያዩ መልክ ሲቀርብ ዜና ፣ ዘገባ ፣ ወይም ኢንፎርሜሽን (information) ብለን እንጠራዋለን። ይታና ኢንፎርሜሽን ሁለት የተለያዩ ነገሮች ናቸው።

ይታ 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	

ይታዎች አይነት አላቸው። ተግባራቸውና የሚይዙት ቦታ አይነታቸውን ይወስናል። ባጠቃላይ ረገድ ፣ አፈራረጃቸው እንደሚከተለው ነው።

1. የቀሩዋር ይታ አይነታት
2. የንባብ ይታ አይነት
3. የቡሉኤን ይታ አይነት
4. የመደብ ይታ አይነታት ናቸው።

የጃቫ ይታ አይነታት በሁለት ይከፈላሉ።

1. መሠረታዊ የይታ አይነታት

አይነታት	መጠበቅ የሚችሉት
byte	n -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 የማይሻግ ነው። መለወጥ ወይም በሽራፊ መቀየር በጭራሽ ክልክል ነው ፤ ስማቸውን ለተለየ ነገር እስከመጠቀም ድረስ፤ ቀጥለን የመሠረታዊ ዴታ አይነታትን አጠርና ዘርዘር በሆነ አቀራረብ እንመለከታለን።

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

byte, char, short, int, long

ቁጥሮች ስለሚደመሩ ፣ ስለሚቀነሱ ፣ ስለሚባዙ ፣ ስለሚካፈሉ እንዲሁም በሌሎች ሥልጣኖች ስለሚገቡ ይህ ጠባይ ልዩ ያደርጋቸዋል።

የchar ዴታ የዩኒኮድ ሆኔ እየተባለ ይጠራል። ምንም እንኳን የድፍን ቀጥር ቤተሰብ አባል ቢሆንም አብይ ተግባሩ ሆኔያትን መወከል ብቻ ነው። በዩኒኮድ ደንብ መሠረት ፊደላት ከ0 በላይ እንጂ ፣ በታች ከድ አይሰየሙም። በመሆኑም አሉታዊ ዕቤት ለchar መሰየም በፍጹም የተከለከለ ነው።

የኢትዮጵያ ፊደል ዩኒኮድ 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 የቀጥር ቤተሰቦች (floating point) ድፍን ቀጥር ብቻ ሳይሆን የተሸረፈው መደዝ ይችላሉ።

በቀላሉ የቀረቡት ክፍሎች የቀረቡት ጋር አንድ አይነት ጠባይ ሲኖራቸው በሚይዙት የዕቅድ አይነት ግን ይለያሉ። ጃቫ ሁለት የተግባራዊ የቀረቡት ይታዩ አይነት አሉት።

`float, double`

- ቡሌኒያን (boolean) ሁለት አቋሞች አሉት፡- አውነትና ሐሰት። እነዚህ ሁኔታዎች በተለያዩ አባባል ሲገለጹ ይሰማል። ለምሳሌ «አለ ፣ የለም» ፤ «አንድ ፣ ዘሮ» ወይም «በራ ፣ ጠፋ»።

`boolean`

3.3 መደባዊ ዴታ

መደባዊ ዴታ ማለት የመደብ ዘር ነው። እያንዳንዱ መደብ የራሱ የሆነ ማንነት አለው። ራሱን ከሌሎች መደባት ጋር ማዘመድ ይቻላል። ካደረገ ፣ የተዘመደውን መደብ ማንነት በራሱ ላይ ይጨምራል።

መደብ ከፈጠርን በኋላ 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 የገድ የመደብ ርቢ (`object`) መፈጠር አለበት። ለምሳሌ የጃቫ እስትሪንግ መደብ ከንባብ ጋር የተያያዙ ችግሮችን መፍትሔ ለመስጠት የምንጠቀመው መደብ ነው። በሥራ ላይ ለማዋል ግን የገድ ርቢ ማውጣት አለብን። በእርግጥ ያለመደባዊ ርቢ የእስትሪንግ 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 ነጻብራቅ ነው።

3.4 ተውላጠ-ቃላት (variables)

ለማንኛውም የኳድራቲክ ጥያቄ አጠቃላይ መፍትሔ ማግኘት ፎርሙላ የሚከተለው ነው።

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

ይህ ፎርሙላ ለማንኛውም ተግባራዊ ቀላላት ያገለግላል። ምክንያቱም ሦስቱ ምልክቶች «**a**» ፣ «**b**» ፣ እና «**c**» ልዩ ልዩ ዕቅዶችን

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

ይኸን ፎርመላ በሥራ ላይ ያዋለ የጃቫ ፕሮግራም እንጻፍ ብንል ፣ የግድ ሦስቱን ዕሴቶች ተቀብለን የምንጠብቅባቸውና ስሌቱን የምንፈጽምበት ተውላጠ-ቃላት ያስፈልጉናል። በተጨማሪ ፎርመላው የሚሰጠውን ሁለት ውጤቶች የምናስቀምጥበት ሌሎች ተውላጠ-ቃላት ማክል 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 ፣ የኢንትጀር ዴታ አይነት ለማለት ነው። በተጨማሪ የኢንትጀር ዴታ አይነት ለት 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 ክፍል (memory) ይከፈታላቸዋል ፤ ነገር ግን ይህ ለመደባወል 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 ድፍነ ቀጥሮችን መጻፍ ወይም ማንበብ ፣ መሠረዝና ልዩ ልዩ የሂሳብ ተግባራትን መፈጸም ይቻላል።

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

3.4.1 የመሠረታዊ ዴታ ተውላጠ-ቃላት አወጣጥ

- `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 = 'u';
```

የ«`fidel`» ተውላጠ-ቃል መሸከም የሚችለው አንድ ሆኖ ብቻ ነው። የያዘውን ሆኖ 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](#)

- `int` ተውላጠ-ቃል

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

ሦስት ተውላጠ-ቃላት ይፈጠራሉ። ባንድ መስመር ወይም በተለያዩ 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 ሲያውጡ መነሻ ዕቅድ መሰይም ይመርጣሉ። ተግባሩን በእንግሊዘኛ initialization ብለው ይጠሩታል።

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

መነሻ ዕቅድ ዘሮ ከሆነ በጃቫ ረገድ ምንም ፋይዳ የለውም። ምክንያቱም ጃቫ ተውላጠ-ቃላት ለሥራ መከሰማራቸው በፊት ራሱ መነሻ ዕቅድ ስለሚሰይማቸው።

- የdouble ተውላጠ-ቃል

```
double ቁጥር = 2.5 ;
```

የደታው አይነት «double» ፣ የተውላጠ-ቃሉ ስም «ቁጥር» እንዲሁም መነሻ ዕቅድ 2.5። የdouble ተውላጠ-ቃል መጠን 8 ባይቶች ነው። ድፍን ወይም ዲሲማል ቁጥሮችን መሸከም ይችላል።

3.4.2 የመደባዊ ተውላጠ-ቃላት አወጣጥ

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

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

በዚህ ውስጥ «fibonacci» የሚለው ተውላጠ-ቃል የFibonacciSequence ርቢ ነው ፤ መደብ ተግባራዊነቱን ለመግለጽ በቅድሚያ ከሚጠይቃቸው ሂደቶች መካከል አንዱ። ይኸንን ቃል በመንተራስ የsequence() መላ በመጥራት የፊልናቺን ተከታታይ ቁጥሮች ማውጣት እንችላለን።

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

በዚህ ቃል መሠረት የተጠራው ተግባር በሥሩን የመጀመሪያ የፊልናቺን ቁጥሮች ያወጣል። ይኸ መሠረታዊ የመደባዊ ተውላጠ-ቃል አቋቋምና አሠራር ነው። ለማክል ያህል ፣ መደባዊ ተውላጠ-ቃል ያለይዘት መፍጠር ይፈቀዳል። በጃቫ ሕግ አይከለክለም። ያለይዘት ፈጥሮ እንዳሥራለንቱ 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 እንዳሥራለንቱ በየጊዜው ይዘቱን መቀያየር ይቻላል። ሌላኛው ፣ ዝምድና ያላቸውን መደባት ለማገናኘትና ለመጠቀም ይረዳል። ይኸን የመጨረሻውን ወደፊት እንደርስባለን።

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

የፈቢናቺ ቀጥሮች፡- 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