መደባዊ ዝምድና (Class Relationship)

8.1 ምዕላደ-ቃላት

አማርኛ ቃል	እንግሊዘኛ ቃል
ውርስ	inheritance
ተወራሽ መደብ	superclass
ወራሽ መደብ	subclass
ተጨባጭ መደብ	concrete class
ረቂቅ መደብ	abstract class
ኧንተርፌስ	interface
የመላ ሽረት	methods overriding
ሞክሼ መላዎች	methods overloading
የመላ ራስጌ	method header
ኤ/ፒ/አይ	API
ኮምፓይል	compile

8.2 መግቢያ

በፕሮግራም ዓለም ውስጥ ጊዜ ከምናባክንባቸው አንዱ ፥ ለተናጠል መፍትሔዎች በተደጋጋሚ የምንጽፈው ኮድ ነው። ይህ የጠና ችግር ከመሆኑ የተነሳ አያሌ መፍትሔ ነክ መንገዶች በየጊዜው ይቀርባሉ። በመደባዊ የፕሮግራም ቋንቋዎች በኩል ይኸን ችግር ለመቀነስ ይረዳ ዘንድ አንድ መደብ ሌላውን ወርሶ የባለቤትነት መብት እንዲኖረው ይፈቅዳሉ።

በአሁኑ ጊዜ ፥ ተግባራዊ ፕሮግራም እንጻፍ ካልን ፥ በቀጥታም ሆነ በተዘዋዋሪ መንገድ ፥ በሌሎች ፕሮግራሞች ላይ መመካቱ አይቀርም። በአያሌ ሁኔታዎች ውስጥ ፥ ከውጭ የሚመጡ መደቦችን እንድንወርስ እንገደዳለን። ለዚህ በቂ ምሳሌ ይሆን ዘንድ የጃቫ ሰርቭለት ኤ/ፒ/አይ **(Java Servlet API)** እንጥቀስ። ይህ ኤ/ፒ/አይ ለዌብ (web) በቀላል መንገድ ፕሮግራም የምንጽፍበት አካባቢ ነው። በግል እንጻፍ ብንል ወራት ወይም ዓመታት ሊወስዱብን የሚችሉ ሥራዎችን በነፃ ይለግሰናል። ሌሎችም ምክንያቶችም አሉ። ኤ/ፒ/አይ ራሱን የሚለግሰው ተጠቃሚው መደብ የ**HttpServlet** መደብን እንዲወርስ በማስገደድ ነው። ራሱን ተወራሽ በማድረግ ከተጠቃሚው መደብ ጋር ባለበቴነትን እንደመጋራት።

ከዚህ የምንገነዘበው ፥ በአንዳንድ ሁኔታዎች ውስጥ የተሰጠን መደብ ሳንወርስ በቀላሉ ፕሮግራም መጻፍ እንደማንችል ነው። የመደብ ዝምድና ጥቅም በዚህ ብቻ አይወሰንም። በቡድን ፕሮግራሞችን ለመጻፍ ቀጥሎ ለመገንባት ማለትም ንድፎችን በመልክ ከፋፍሎ ፥ ድርሻዎችን ለያንዳንዱ የቡድን አባል ደልድሎ ፥ በተናጠል ኮዶችን ጽፎና ፈትኖ ፥ በመጨረሻ ሁሉንም በአንድ አደራጅቶ የመጨራሻውን ፕሮግራም ለማውጣት እጅግ አመቺ ነው።

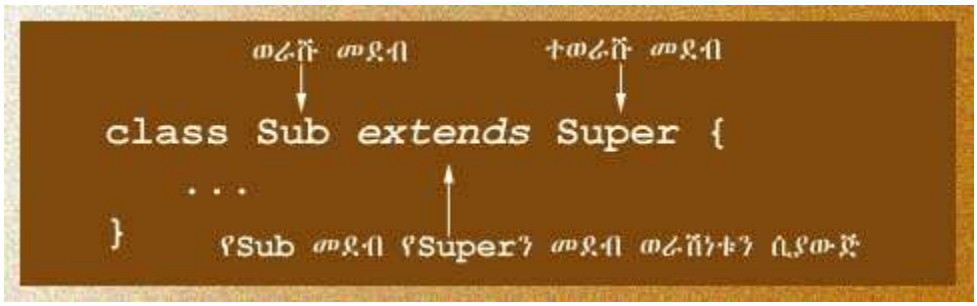
8.3 መደባትን በውርስ ማዛመድ

በሁለት መደቦች መካከል ዝምድና ከምንፈጥርባቸው መንገዶች አንዱ ውርስ ነው። በዚህ ዝምድና አንደኛው መደብ ወራሽ ፥

ሌላኛው አውራሽ ይሆናሉ። ወራሹ መደብ ሁልጊዜ ዝምድናውን የመመሥረት ኃላፊነት አለበት። በሂደቱ ፥ ተወራሹ መደብ ምንም ዓይነት እርምጃ መውሰድ የለበትም።

ዝምድናው ፥ ወራሽ መደብን የተወራሹን ተውላጠ-ቃላት ፥ መላዎች ፥ እንዲሁም ውስጣዊ መደቦች በቀጥታ ማየትና እንደራሱ አድርጐ የመጠቀም መብት ይሰጠዋል። ተወራሹ የግል **(private)** ብሎ የሰየማቸው አባላት ግን በፍጹም ማየት ወይም መንካት አይችልም። በprivate ፍቃድ ሥር ያሉ የመደብ አባላት ከባለቤታቸው ክልል ውጭ አይታዩም። ለወራሽ መደብ ጨምሮ።

አንድ መደብ ሌላ መደብ እንዲወርስ ከተፈለገ ፥ በመደቡ ድንጋጌ ጊዜ የ**extends**ን ቃል በመጠቀም ዝምድናውን እንመሠርታለን። አጻጻፉ እነሆ፦



ቀጥለን የውርስ ዝምድናን አጻጻፍ በተግባራዊ ምሳሌ እንመለከታለን። ሁለት መደቦች አሉ፦ **Super** እና **Sub**። ወራሹ Sub ሲሆን ተወራሹ ደግሞ Super ነው። የSubን መደብ በጥንቃቄ ከተመለከትን ፥ የት ቦታ ላይ ራሱን ከተወራሹ መደብ ጋር እንደሚያዛምድ እናገኛለን። ዝርዝሩን ከኮዱ በታች እንጐብኝ።

ወራሽና ተወራሽ መደቦች

```
/** Class: Super  */
class Super {
   double field = 0 ;

   /* Sets the field variable a value */
   void set(double value) {
      field = value ;
   }

   /* Returns the value of field */
   double get() {
      return field ;
   }
}

/** Class: Sub  */
public class Sub extends Super  {
   /** Method: An entry point for program execution */
   public static void main(String[] args) {
      Sub s = new Sub() ;
      s.set(3.14159) ;
      System.out.println("field = " + s.get()) ;
   }
}
```

Download: Sub.java

ይህ ምሳሌ ፥ በተወሰነ ደረጃ ቢሆንም አንድ መደብ ሌላውን እንዴት እንደሚወርስና እንደሚጠቀም ያሳያል። የደመቁትን ቃሎች በተርታ እናብራራ።

- የSub መደብ ራሱን ከተወራሹ መደብ ጋር ያዛመደው በ**extends** ቃል ነው። ይኸን ብቻ በማድረግ ፥ የተወራሹን መደብ አካሎች ብቻ ሳይሆን በተጨማሪ ተወራሹ መደብ ሲወርድ ሲዋርድ የወረሳቸውን መደቦች በሙሉ ይወርሳል። ይሁን እንጂ ፥ በአንድ ወቅት ፥ ማንኛውም መደብ በቀጥታ መፍጠር የሚችለው የውርስ ዝምድና ከአንድ ተናጠል መደብ ጋር ብቻ ነው።
- የSub የመደብ ርቢ ፥ ከተወራሹ መደብ የመጡት መላዎችን እንደራሱ ሰለሚያያቸው (መሆን አለበት) በቀጥታ ይጠራል ፤ ይጠቀማል። ሁለቱ የመላ ጥሪዎች **s.set(3.14159)** እና **s.get()** ይኸንኑ ያሳያሉ።
- ሁለት መደቦች የውርስ ዝምድና ለመፍጠር በግድ አንድ ፋይል ውስጥ መኖር የለባቸውም። በአንድ ፋይል ፥ ፓኬጅ ፥ ጃር ፥ ወይም በየቦታው መኖር ይችላሉ። ዋናው ነገር መተያየት የሚያስችላው ጐራ መኖሩ ነው።

8.4 ሁሉም መደቦች ወራሽ ናቸው

ወደደም ጠላም ፥ ማንኛውም የጃቫ መደብ ፥ ቢያንስ አንድ መደብ የመውረስ ግዳጅ አለበት። ስለዚህ ማንኛውም መደብ ወራሽ ነው።

ይህ አባባል ጥብቅ ጥያቄ ያስነሳል። እስካሁን ያየናቸው ምሳሌዎች (ከላይኛው በስተቀር) ሌላ መደብ ሲወርሱ ወይም ሙከራ ሲያደርጉ አላየንም። በአንድ በኩል ማንኛውም መደብ ወራሽ ነው እያልን ፥ በሌላ በኩል ደግም ያንን ቃል የሚሸር ምሳሌዎች ስንመለከት ሰንብተናል። በእርግጥ ቅራኔ አለ ወይስ የማናውቀው ምስጢር?

አንድ መደብ ሌላ ሳይወርስ ራሱን መደንገጉ ከታወቀ ፥ የጃቫ ኮምፓይለር በራሱ የመደቡን ድንጋጌ በማደስ ከ**Object** መደብ ጋራ የወራሽና የተወራሽ ዝምድና ይፈጥራል። ድርጊቱ የሚፈጸመው በኮምፓይሌሽን ወቅት ነው። በምዕራፍ ፪ የቀረበ ምሳሌ ኮምፓይል ከመደረጉ በፊትና በኃላ እርቃነ-ሥጋውን ምን እንደሚመስል ተከታዩ ቃል ያሳያል።

ወራሽነቱን በራሱ የማያስቀምጥ መደብ፦

```
public class UnicodeChar {
   ...
}
```

ወራሽ መደብ እንዲሆን በጃቫ ከተገደደ በኃላ፦

```
public class UnicodeChar extends java.lang.Object {
   ...
}
```

በጃቫ ሕግ መሠረት ፥ ቢያንስ አንድ መደብ የ**Object**ን መደብ የመውረስ ግዳጅ አለበት። በመሆኑም ፥ የObject መደብ ለሁሉም ተወራሽ መደብ ነው። በቀጥታ ወይም በተዘዋዋሪ እሱን የማይወርስ ወይም የማይዛመድ መደብ የለም። ይህ ዝምድና እስካሁን ድረስ ያልተጠቀሰ ተጨማሪ ጥቅም አለው። የObject ተውላጠ-ቃል የማንኛውንም የመደብ ርቢ የማዘል ችሎታ

አለው። ለምሳሌ ፥ ልዩ ልዩ የመደብ ርቢዎችን ሊጠብቅ የሚችል ኧሬይ መፍጠር የምንሻ ከሆነ የObjectን መደብ መጠቀም አለብን።

```
Object[] misc = new Object[11] ;
misc[0] = new String("Addis Ababa") ;
misc[1] = new Double(2.71) ;
misc[2] = new UnicodeChar() ; ...
```

ይህ ባይሆን ኖሮ ለእያንዳንዱ መደብ የራሱን አምሳል ኧሬይ መፍጠር እንገደድ ነበር። በሰፊው ከሚታወቁት የዴታ ማደረጃ ስልቶች መካከል አንዱ «ኧስታክ» **(Stack)** ሲሆን የObject መደብን ልማትና አጠቃቀም በትንሹ ለማሳየት ይረዳ ዘንድ እዚህ በምሳሌነት ቀርቧል። ይህ ባለኧስታክ (Stack) ኮድ አንባቢው ሌላ ቦታ ካየው ከተለየ ግር ሊሰኝ አይገባም ፤ አነስተኛ ልዩነት እዚህም እዚያ ይኖራልና።

ሁሉም መደቦች ወራሽ መደብ ናቸው

```
/** Class: Stack */
import java.util.* ;
class Stack {
   final int size = 16 ;
   Object[] store = new Object[size] ;
   int top = -1 ;    // where the stack index is

   /* Writes data into the stack */
   void push(Object element) {
      if (top < size-1)
         store[++top] = element ;
   }

   /* Returns the top most element from the stack */
   Object pop() {
      if (top < 0) return null ;
      return store[top--] ;
   }

   /* Returns true if the stack is full */
   boolean isFull() {
      if (top < size-1) return false ;
      return true ;
   }

   /* Returns true if the stack is empty */
   boolean isEmpty() {
      if (top >= 0) return false ;
      return true ;
   }
}

/** Class: StackDemo  */
public class StackDemo extends Stack  {

   /** Method: An entry point for program execution */
   public static void main(String[] args) {
      Random rand = new Random() ;
      StackDemo sr = new StackDemo() ;
```

```
        // Fill the stack with whole numbers
        while( !sr.isFull())
           sr.push(new Integer(rand.nextInt(1024))) ;

        // Pop and print the stack contents
        while (!sr.isEmpty())
           System.out.println("Pop -->" + (Integer) sr.pop()) ;
    }
}
```

Download: StackDemo.java

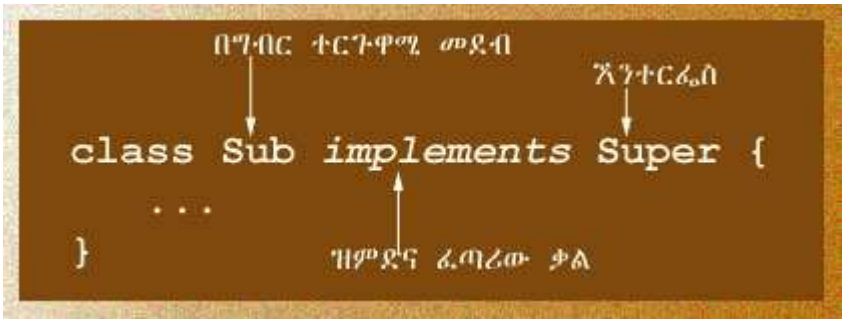
8.5 ጃቫ ኧንተርፌስ (Java Interface)

የጃቫ ኧንተርፌሶች ከመደብ ጋር ይመሳሰላሉ ፥ ነገር ግን ማቀፍ በሚችሏቸው አባላት ጥብቅ መሠረታዊ ይለያሉ። በራሳቸው የመቆም ወይም የመንቀሳቀስ ብቃት የላቸውም።

አባሎቻቸው ፥ የመላ አዋጆች ፥ ቋሚና ኧስታቲክ ተውላጠ-ቃላት ናቸው። የመላ አዋጅ ስንል ፥ ድንጋጌ የሌለው ፥ በድርብ-ሰረዝ (;) የተቋጨ የመላ ራስጌ ማለታችን ነው። አባሎቻቸው ሁልጊዜ ገሀድና ለማንኛውም የውጭ መደብ ሆነ ኧንተርፌስ ክፍት ናቸው።

ኧንተርፌሶች ለዝምድና ሁልጊዜ ዝግጁዎች ናቸው። ተፈጥሯዊ ባሕሪያቸው እንዲሉ። ሌላ መደብ ወስዶ ተግባራዊ ካላደረጋቸው ፥ በፍጹም ዋጋ የላቸውም። አብይ ዓላማቸው ፥ በማይተዋወቁ መደቦች ፥ እንዲሁም ፕሮግራሞች መካከል ድልድይ መፍጠር ነው። የጃቫ ኤ/ፒ/አይ ለዚህ በከፍተኛ ደረጃ ይገለገላቸዋል።

የኧንተርፌስ ዝምድና ቃል አገባብ፦



የኧንተርፌስ አደነጋገና አጠቃቀም ምሳሌ፦

ኧንተርፌሱና የመደብ ዝምድና

```
/** Interface: Instrument  */
interface Instrument {
   public void play() ;
}

/** Class: Piano  */
```

```
public class Piano implements Instrument  {
   /* Returns the value of field */
   public void play() {
      System.out.println("Playing: P I A N O") ;
   }

   /** Method: An entry point for program execution */
   public static void main(String[] args) {
      Piano piano = new Piano() ;
      piano.play() ;
   }
}
```

Download: Piano.java

ይህ ኮድ አንድ ኧንተርፌስና መደብ አሉት። በሁለቱ አካሎች መካከል ዘምድና አለ። የ**Piano** መደብ የ**implements**ን ቃል በመጠቀም እሱ ትርጒም አቅራቢ ፥ የ**Instrument** ኧንተርፌስ ግን ተተርጓሚ ሆነው ዝምድናው ይመሠረታል። በዚህ ዝምድና ወይም ግንኙነት ፥ የPiano መደብ በኧንተርፌሱ ውስጥ የታወጁትን መላዎች በሙሉ የመደንገግ ወይም ሙሉ ሕይወት የመስጠት ግዳጅ አለበት።

ማንኛውም መደብ ከአንድ ወይም ከብዙ ኧንተርፌሶች ጋር በቀጥታ መዛመድ ይችላል። ይህ ፍቃዳዊ ነው። ግዳጅ የለበትም። በአንጻሩ ፥ መደቦች ግን በቀጥታ ከአንድ መደብ በላይ መውረስ ወይም ራሳቸውን ማዛመድ አይፈቀድላቸውም።

◀ ተመለሥ ማውጫ ቀጥል ▶

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