ስሕተቶችና አስተራረማቸው

9.1 ምዕላደ-ቃላት

አማርኛ ቃል	እንግሊዘኛ ቃል
ንባባዊ ፕሮግራም	source code
ስህተት	error
ግድፈት	bug
ቅድመ-ስምሪታዊ ስህተት	compile time error
ስምሪታዊ ስህተት	runtime error
እርማት	debugging
ግድፈት ማጣሪያ	debuger
ኧክሴፕሽን	Exception
ውዴታዊ ኧክሴፕሽን	unchecked exception
ግዴታዊ ኧክሴፕሽን	checked exception
ኧክሴፕሽን አያያዝ	exception handling

9.2 ስህተቶች

ተቀበልነውም አልተቀበልነውም ፥ ፕሮግራም ስንጽፍ ስህተቶች እንሠራለን። ይኸን የሚቃወም ሰው ካለ ራሱ ስህተት መሆን አለበት። ምክንያቱም ፥ አንድ ፕሮግራም ከስህተት ወይም ከግድፈት መቶ በመቶ ነጻ መሆኑን ማረጋገጫ መንገድ የለም። የኮምፕዩተር ፕሮግራም ስህተቶች ከድርሰት ጽሑፍ ስህተቶች ጋር ይመሳሰላሉ። የአንድ ጽሑፍ ዐርፍተ-ነገር የሰዋሰውን ሕግ ከጣሰ ፥ ትርጒሙ ይዛባል። እንዲሁም ከመዝገበ-ቃላት ውጭ ወይም ተስምቶ የማይታወቅ ቃል ቢያስገባ ጽሑፉ የታሰበውን ሀሳብ ከመግለጽ ይሰናከል ይሆናል። አብዛኛውን ጊዜ ግን ፥ አንባቢያን ሁኔታዎችን በማመዛዘን የጽሑፉን አባበል ገምተው ትክክለኛው ነጥብ ላይ ሊደርሱ ይችላሉ። የፕሮግራም ስህተቶች ግን አይምሬ ናቸው። ቢያሻ መታረም አለባቸው ፥ አለዛ እንደጥልቀታቸው አበሳ ይፈጥራሉ። ማንነታቸው ከታወቀ ፥ ወዲያውኑ መታረም አለባቸው።

በዚህ ጥናት ውስጥ «ስህተት» ወይም «ግድፈት» ስንል የኮምፕዩተር ስህተት ወይም ግድፈት ለማለት ነው። የስህተት ምንጮች ፥ የጃቫን የሰዋሰው ሕግ የጣሱ የቃል አገባቦች ፥ ግድፈትን ያቀፉ ተውላጠ-ቃላት ፥ ጠንቅ ያዘሉ የኮድ ስልቶች ፥ እንዲሁም የምንጠቀማቸው መሣሪያዎች ናቸው።

ስህተቶች መቀነሻ አስከትሎ ማረሚያ መንገዶች አሉ። ዋናውና የመጀመሪያው የጸሓፊው አሠራር ፥ ጥንቃቄና ሥነ-ሥርዓት ነው። አንድ ኮድ ተጽፎ ለኮምፓይሌሽን ሂደት ከመድረሱ በፊት በወረቀት ላይ በቅደም ተከተል ተመርምሮ ፥ ውጤቱ ተጣርቶ ፥ መሥራትና አለመሥራቱ መረጋገጥ አለበት። ይህ ደረጃ ፥ ልዩ ልዩ ስህተቶችን ለማግኘትና ተገቢውን እርምጃ ለመውሰድ ዕድል ይሰጣል። ከፋቱ ፥ ብዙ ፕሮግራም ጸሐፊዎች ይኸን መንገድ መከተላቸው አጠራጣሪ ነው።

ሌላኛው መንገድ ፥ የጃቫ ኮምፓይለር አንድ ኮድ ሲመለከት ስህተት ካገኘ እንዲታረም ሲጠይቅ ነው። የጃቫን ሕግ የጣሰ ማንኛውም ንባባዊ ፕሮግራም ለስምሪት የሚያስችለው አካል ላይ እንዲደርስ አይፈቀድለትም። ስህተቶቹን እስኪያርም ድረስ።

ለፕሮግራምነት ከበቃም በኃላ የተጠበቀውን ያህል መሥራትና አለመሥራቱን የማጣራቱ ሂደት እጅግ አስቸጋሪና እርግጠኝነት የጐደለው ነው። ስለዚህ ፕሮግራሙ ከተለያዩ ማእዘኖች አንጻር በጥብቅ መፈተን አለበት። ፕሮግራሙ ስምሪት ላይ እያለ ደረጃ በደረጃ ውስጣዊ ሥራውን መከታተልና የችግሮችን መንስኤ ማፈላለግ እንዲሁም ማረም ይቻላል። የጃቫ ፕሮግራም መገንቢያ አብሮት የሚመጣ «ግድፈት ማረሚያ» **(debugger)** አለው።

የመጨረሻ እንኳን ባይሆን ሌላኛው ስህተት መፈለጊያና ማረሚያው መንገድ ተጠቃሚው ሕዝብ ነው። በአሁኑ ጊዜ አንድ ፕሮግራም አለቀ ተብሎ በገሀድ ከመሠራጨቱ በፊት በተጠቃሚዎች እንዲሞከር ማድረጉ የታመነበትና ቢቻል በሥራ የሚውል ነው።

9.3 ቅድመ-ስምሪታዊ ስህተቶችና አስተራረማቸው

በቅድመ-ስምሪታዊ ስህተቶች የተበከለ አንድ ንባባዊ-ፕሮግራም ለግንባታ ዝግጁ አይሆንም። ስህተቶች የግድ በመጀመሪያ መታረም አለባቸው። ባለቅድመ-ስምሪታዊ ስህተቶች ፥ አብዛኛውን ጊዜ ፥ የተሳሳተ ቃል አገባብ ፥ የተረሳ ወይም የተዛነፈ ሥርዓተ-ነጥብ ፥ ወይም ባለግድፈት ተውላጠ-ቃል ናቸው።

ቀላል ይመስላሉ ፤ ግን ዕለታዊ ችግሮች ናቸው። ለምሳሌ ያህል ፥ x እና y ቀደም ብለው እንደታወጁ እንውሰድና ይኸን ቃል እንመልከት።

```
x =+ y ;
```

ይህ ምስኪን መስሎ የሚታይ ቃል በማይታለፍ ስህተት ተገድፏል። እርማት ሳይደረግ ለጃቫ ኮምፓይለር ከቀረበ በፍጹም ሂደቱን አያልፍም። ስህተቱ የሂሳብ ምልክት አሰካኩ ላይ ሲሆን ትክክለኛው ቃል ይኸን ይመስላል።

```
x += y ;
```

ሥርዓተ-ነጠብ መሳት ወይም አላግባብ መጠቀም እንዲሁ የከፋ የስህተቶች መንስኤ ነው። ስንቶቻችን ነን የሚከተለውን ዓይነት ስህተት ደግመን ደጋግመን የምንሠራው?

ስህተት ያዘለ መደብ

```
public class Temari {

   public stataic void main(String[] args) {
      ...
   }
   }
}
```

ብዙ ነን ቢባል ከእውነት አይርቅም። ለዚህ መድኃኒቱ የሚከተሉት ናቸው።

- ንባባዊው ፕሮግራም በትክክል መሥራትና አለመሥራቱን በወረከቀት ላይ በጥንቃቄ በድጋጋሚ ማጣራት። ይኸን ሂደት በእንግሊዘኛ «**tracing**» ይሉታል።
- ንባባዊው ፕሮግራም ኮምፓይል ሲደረግ ስህተት ከተገኘ ፥ ስልስህተቱ የተሰጠውን መልክት በጥንቃቄ ማንበብና የስህተቱን እውነተኛ ምንጭ ማፈላለግ። አንዳንድ ጊዜ ኮምፓይለሩ ስህተቱ ተነሰ ያለበት መስመርና እውነተኛው የስህተቱ አመንጪ መስመር አንድ ላይ ይሆናል ይችላል። በጣም አሥፈላጊ ነጥብ።
- የስህተቶቹ ቁጥር ብዙ ከሆነ ፥ የተወሰነውን ንባባዊው ፕሮግራም ለጊዜው ወደ ትችት መቀየር። ይህ ደረጃ በደረጃ

ስህተቶችን ማጥቂያ መንገድ ይከፍታል።

ከዚህ በታች የተሰጠው ምሳሌ ብርቱ ስህተት አለው። የማረሙን ሥራ ለአንባቢው ትተን ወደ ቀጣዩ ክፍል እናመራለን።

ስህተት ያዘለ መደብ

```
import java.util.* ;
public class Finderror {

   /** Method: An entry point for program execution */
   public static void main(String[] args)  {
      System.out.println("Today is " + new Date()) ;
   }
}
```

Download: FindError.java

9.4 ኧክሴፕሽን ፥ የስህተቶች መቆጣጠሪያ ፥ መከላከያ

ከላይ እንዳየነው ፥ ቅድመ-ስምሪታዊ ስህተቶች ካልታረሙ በስተቀር ወደፊት መንቀሳቀስ አለመቻሉ ግልጽ ነው። የጃቫ ኮምፓይለር ማቆም የሚሳነው የስህተቶች ዓይነትና ቁጥር ብዙ ነው። አንድ ፕሮግራም የኮምፓይል ደረጃ ማለፉ ከስህተት ነፃ አያደርገውም ምንም እንኳን ያ አንድ እርምጃ ቢሆንም። ፕሮግራሙን የመቆፈሩና ስህተቶቹን የማረሙ ሂደት መቀጠል አለበት።

በጃቫ ይነሳሉ ብለን የምናስባቸውን ስህተቶች ከመሆናቸው በፊት የምናገኝበትና እርምጃ የምንወስድበት መንገድ አለ። ተጠቃሚውን ቁጥር አስገባ ብለነው ከቁጥር ፈንታ ኧስትሪንግ ቢያስገባ ምንድን ነው የምናደርገው? የኧስትሪንግ ዴታ የቁጥር ቦታ ውስጥ ማስገባት አይፈቀድልንም። ከሞከርን «ስምሪታዊ ስህተት» ይፈጠራል። የፕሮግራማችን ሂደት ይሰናከላል። እንደዚህ ዓይነት ስህተቶችን ለማቆም መፍትሔው ኧክሴፕሽን ነው።

በተጨማሪ ፥ የጃቫን መደቦች ስንጠቀም ስህተት ከተነሳ ችግሩን የሚያስታውቁን ከስህተቱ ጋር የተዛመደውን ኧክሴፕሽን ወደእኛ በመወርወር ነው። የተወረወረውን የኧክሴፕሽን ርቢ ተቀብለን የስህተቱ ዓይነት ካወቅን በኃላ ተገቢውን እርምጃ እንወስዳለን። በመሆኑም ፥ የኧክሴፕሽን ስልት ስምሪታዊ ስህተቶችን ለመከታተልና እርምጃ ለመውሰድ ዕድል ይሰጠናል። የቃል አገባቡ ይኸን ይመስላል።

ይህ ምስል የሚያሳየው ፥ አንድ ኧክሴፕሽን ሲወረወር እንዴት መያዝ እንዳለበት ነው። ኧክሴፕሽን ወርዋሪና ተቀባዮች ሁልጊዜ የመደብ አባላት ሜላዎች ናቸው። የጃቫን መደብ ሰንጠቀም ማለትም ርቢ ፈጥረን መላዎቻቸውን ስንጠራ ስህተት ከተፈጠረ ፥ ከስህተቱ ጋር የሚዛመድ ኧክሴፕሽን ወደእኛ ይወረውራሉ። የ**try** ክፍል ማቀፍ ያለበት የመላ ጥሪዎችንና የመሳሰሉትን ነው። የ**catch** ክፍል እንደ ኧክሴፕሽኑ ዓይነት ስህተት አራሚ ወሳጅ ነው። አንድ ወይም ከአንድ በላይ የ**catch** ክፍል ይፈቀዳል። የ**finally** ክፍል ኧክሴፕሽን ቢወረወር ባይወረወርም በሥራ ላይ ይውላል። አማራጭ ስለሆነ የግድ መግባት የለበትም። የጸሐፊው ፈንታ ነው።

ቀጣዩ ምሳሌ የጃቫ Double መደብ አባል የሆነውን አንዱን መላ በኧስትሪንግ መልክ የላከውን ዴታ ወደ **double** የዴታ ዓይነት እንዲቀይር ይጠራል። ያ መላ የተላከለት የኧስትሪንግ ዴታ ችግር ካለበት ወደ ጠሪው የ**NumberFormatException** ይወረውራል። አያያዙ ምን እንደሚመስል በሜይን መላ ውስጥ ቀርቧል። አንባቢው

ምሳሌውን ገንብቶ ውጤቱን ካየ በኃላ ፥ ለ**values** የተሰየመውን ዕሴት አንዱን ገድፎ ፤ ማለት **2.71** ወደ **2.71A** ቀይሮ እንደገና ፕሮግራሙን ገንብቶ ለሥራ ቢያሰማራው ፥ ተወርዋሪ ኧክሴፕሽኑና ማንነት በግልጽ ይመለከታል።

ኧክሴፕሽን አያያዝ

```
public class Filter {

   /** Reads double type from keyboard */
   double toDouble(String stringDouble)  {
      double d = Double.parseDouble(stringDouble) ;
      return d ;
   }

   /** Method: An entry point for program execution */
   public static void main(String[] args)  {
      String[] values = {"2.71", "3.14159", "1.6" } ;
      Filter f = new Filter() ;

      // handles NumberFormatException
      try {
         for (int i=0; i < values.length; i++)
            System.out.println("v = " + f.toDouble( values[i])) ;
      } catch (NumberFormatException e) {
         System.out.println(e) ;
      }
   }
}
```

Download: Filter.java

9.5 የኧክሴፕሽን ዓይነቶች

ሦስት የኧክሴፕሽን ዓይነቶች አሉ ፤ ግን እዚህ በሁለቱ ላይ ብቻ እናተኩራለን። እነሱም ፥ **ውዴታዊ ኧክሴፕሽን (unchecked exception)** እና **ግዴታዊ ኧክሴፕሽን (checked exception)** ናቸው።

ውዴታዊ ኧክሴፕሽኖች **(unchecked exceptions)** ፕሮግራሞች በሥራ ላይ እያሉ የሚመነጩ ፤ ነገር ግን በግዴታ መያዝ ያለባቸው አይደሉም። መላዎች በሥራ ላይ እያሉ ስህተት ተፈጥሮ እንደዚህ ዓይነት ኧክሴፕሽን ከወረወሩ ፥ ጠሪው የመቀበል ወይም ያለመቀበል አማራጭ አለው። ለዚህም ነው «ውዴታዊ ኧክሴፕሽን» የምንላቸው።

ከጃቫ ጋር አብረው የሚመጡ የውዴታዊ ኧክሴፕሽኖች ቁጥር ብዙ ከመሆኑም በላይ ሥፍር ቁጥር የሌላቸው የስህተት ዓይነቶች ይወክላሉ። እንዳስፈላጊነታቸው ፕሮግራማችን ውስጥ ለውርወራም ሆነ ለቀበላ መጠቀም እንችላለን። የጃቫ ኤ/ፒ/አይ በጣም ይገለገላቸውም ፤ እናም እሱን በተጠቀምን ጊዜ ሁሉ ከእነዚህ ኧክሴፕሽኖች ጋር ጋር በተዘዋዋሪ ሆነ በቀጥታ እንገናኛለን። ከውዴታዊ ኧክሴፕሽኖች መካከል በጣም ጥቂቶቹን እንጥቀስ።

NullPointerException

ይህ ስህተት የሚነሳው ኑል **(null)** የሆነ ተውላጠ-ቃል መላዎችን ለመጥራት ሲሞክር ፤ ኑል የሆነን ኧሬይ የሕዋሳትን ቁጥር ለማወቅ ሙከራ ሲደረግና የመሳሰሉት ናቸው። ቀጣዩ ቃል ይኸን ያሳያል።

```
int[] numbers = null ;
numbers[3] = 128 ;
```

አንድን ቁጥር በዚሮ ለማካፈል መሞከር ውጤቱ ቀውስ ነው። በሥነ-ሂሳብ ውስጥ በፍጹም ትርጉም የለውም። እንዳጋጣሚ ፥ ይህ ሁኔታ ከተከሰተ ፥ ጃቫ የሚከተለውን ውዴታዊ ኧክሴፕሽን ይወረውራል።

```
ArithmeticException
```

ሌላኛው የኧክሴፕሽን ዓይነት ግዴታዊ ነው ፤ ማለትም እንደዚህ ዓይነት ኧክሴፕሽን ወርዋሪ መደብ ከተጠቀምን የመቀበል ግዳጅ አለብን። አሻፈረኝ አማራጭ አይደለም። የጃቫ ኮምፓየለር ግዴታዊ ኧክሴፕሽኖች በሥነ-ሥርዓት መያዛቸውን ሳያረጋግጥ አንድን ፕሮግራም ለግንባታ አያስተላልፍም። ነገሩ የውዴታ ግዴታ ነው። ዴታ ከፋይል የምናነብባቸው የጃቫ መደቦች እንደዚህ ዓይነት ኧክሴፕሽን ወርዋሪ ስለሆኑ መቀበሉ አማራጭ የሌለው ተግባር ነው።

በግዴታዊና ውዴታዊ ኧክሴፕሽኖች መካከል ያለው ልዩነት ግዴታዊው በአግባብ መያዝና አለመያዙን የጃቫ ኮምፓይለር ማረጋገጡ ነው። ይኸንን በትንሹ ለመገንዘብ ይረዳን ዘንድ ቀጣዩን ኮድ እንመርምር። ዴታ ወደ ፋይል ለመጻፍ የተጠቀማቸው መዶቦች ኧክሴፕሽን ከወረወሩ አቀባበሉና እርምጃ አወሳሰዱ እንዴት መሆኑን ይጠቀሙማል። የተጠቀሰውን ፋይል መፍጠር ካልተቻለ ፥ በእርግጥ ኧክሴፕሽን ይነሳል።

ግዴታዊ ኧክሴፕሽን አያያዝ

```
import java.io.*; ;
public class Cities {

   /** An entry point for program execution */
   public static void main(String[] args)  {
      String[] names = {"Addis Ababa", "Awassa","Dire Dawa","Axum"} ;

      // handles IO exceptions
      try {
         // creates a file obj representing a file
         File out = new File("cities.txt") ;

         // creates a writer obj
         FileWriter fw = new FileWriter(out) ;

         // writes the strings into the file
         for (int i=0; i < names.length; i++)
            fw.write(names[i] + '\n') ;

         // closes the output stream
         fw.close() ;

      } catch (IOException e) {     // exception handler
         System.out.println("File write failed") ;
      }
   }
}
```

Download: Cities.java

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