ምእራፍ ፪

የጃቫ ፕሮግራም መዋቅርና አቋቋም

የጃቫ ፕሮግራም የሚያሻማና ግልጽ የሆነ መዋቅር አለው። የመዋቅሩ መሠረታዊ ክፍል መደብ (class) ተብሎ ይጠራል። ማንኛውም የጃቫ ፕሮግራም ቢያንስ አንድ ወይም ካንድ በላይ መደቦች ይኖሩታል። ባጠቃላይ አነጋገር የጃቫ ፕሮግራም ስንል ፥ በሥነ ሥርዓት የተዋቀሩ የመደቦች ስብስብ ማለታችን ነው።

2.1 ማብራሪያ (Comments)

ፕሮግራሞች ለተለያዩ ችግሮች መፍትሄ የሚለግሱ ሥራዎች ናቸው። የመመሪያዎቹ ቁጥር እየጨመረ ሲመጣ ፥ እነሱን በቅጡ አንብቦ የመረዳቱ ችግር እየበረታ ይመጣል። ስለዚህ የፕሮግራሞችን ተነባቢነት ለመጨመር ከሚደረጉት ጥረቶች መካከል አንዱ ማብራሪያዎችን ፕሮግራሞቹ ውስጥ መክተት ነው። ለዚህ ፥ ጀቫ ሁለት አይነት ማብራሪያ መጻፊያ መንገዶች አሉት።

1. በሁለት «የጭረት» (//) ምልክቶች የሚጀምሩ መሥመሮች በሙሉ ማብራሪያ ናቸው። ማብራሪያው ካንድ መሥመር በላይ ከሆነ ፥ እያንዳንዱ መሥመር የግድ በሁለት የጭረት (//) ምልክቶች መጀመር አለበት። እንደነዚህ ያሉትን ምልክቶች የ«ማብራሪያ ምልክቶች» ብለን ልንጠራቸው እንችላለን።
2. ሁለተኛው መንገድ ካንድ መሥመር በላይ የሆነ አስተያየት ለመጻፍ አመቺ መንገድ ነው። ማብራሪያው በ«/*» ተከፍቶ በ«*/» ይዘጋል።

የጃቫ ማብራሪያዎችን የትኛውም የፕሮግራማችን ክፍል ውስጥ መጻፍ ይፈቀዳል። ምንም እንኳን ይህ እውን ቢሆንም ማብራሪያዎች የፕሮግራም መመሪያ ቃላት አባላት አይደሉም። ስለሆነም አጻጻፋቸው ሕግን ካልጣሰ በስተቀር የጃቫ ኮምፓይለር አይመለከታቸውም። ምክንያቱም ተግባር እንዲፈጸም የሚጠይቁ መመሪያዎች አይደሉምና።

ማብራሪያ አጻጻፍ

```
01. class Parameters {
02.   /*
03.    * An entry point to execution
04.    * @param args captures command line arguments
05.    */
06.   public static void main(String[] args) {
07.      if (args.length <= 0 ) return ;
08.
09.      // print the  command line arguments
10.      for (int i=0; i < args.length; i++)
11.         System.our.println(i + ". " + args[i]) ;
12.   }
13.}
```

ይህ ምሳሌ ፥ ሁለቱንም የማብራሪያ አጻጻፍ መንገዶች ያሳየናል። ከተራ-ቁጥር 2 እስከ 5 ያለው አንደኛው አይነት ማብራሪያ አጻጻፉር ሲሆን ቁጥር 9 ላይ ያለው ደግሞ ሌላኛው ነው። እነዚህ ማብራሪያዎች ለፕሮግራም ሎጂክ የሚሰጡት ምንም አስታዋጽኦ የለም። ለፕሮግራም ጸሐፊዎች ግን እጅግ ተፈላጊ ናቸው። ምክንያቱም ፕሮግራሙ የሚሠራውን ነገር በቀላሉ ለመግለጽና ለመገንዘብ ይረዳሉ። አስቀድመን እንዳየነው ፥ የጃቫ ኮምፓይለር ማብራሪያዎችን እንደ መመሪያ ቃል አይመለከታቸውም። ማረጋገጥ የሚፈልገው በሥነ ሥርዓት መጻፋቸውን ብቻ ነው።

2.2 መደብና አመዳደብ

ሥራ የሚያፈራ የጃቫ ፕሮግራም ለመገንባት ቢያንስ አንድ መደብ መኖር አለበት። ያለመደብ የሚጻፍ የጃቫ ፕሮግራም የለም። ፕሮግራም ጸሐፊው ውጥኑን ከዘረጋና ፥ ንድፉን ካስቀመጠ በኃላ ፥ መደብ ሲመድብ የጃቫን መዋቅር በንጽህና የመከተል ግዳጅ አለበት።

እያንዳንዱ የጃቫ መደብ ያራሱ የሆነ ክልል አለው። አንዱ ከሌላው እንዲጋጭ አይፈቀድለትም። ግንኙነት መመስረት በሚገባ ይችላል። ነገር ግን በአካል ደረጃ ፥ ሕገ ወጥ በሆነ መንገድ የራሱን ክልል አልፎ የሌላውን መሻማት አይችልም። የራሱን አጥር በሥነ ሥርዓት የማጠር ኃላፊነትም አለበት።

የጃቫ አፈጣጠር ወይም አመዳደብ እንደሚከተለው ነው። ይኸን ኮድ ለማብራራት ይረዳ ዘንድ እያንዳንዱ መሥመር መጀመሪያ ላይ ተራ-ቁጥር ተጨምሯል። የተራ-ቁጥሩ ዓላማ ይህና ይህ ብቻ ስለሆነ ፥ እውነተኛ ፕሮግራሞቻችን ውስጥ በፍጹም መጨመር የለብንም ፥ ብንሻም መጨመር አይፈቀድልንም።

የመደብ አመዳደብ

```
01. class Account {
02.   private double balance = 0 ;
03.
04.   void deposit(double d) {
05.      balance += (d <= 0 ? 0 : d) ;
06.   }
07.
08.   void withdraw(double w)   {
09.      if (w <= balance)
10.         balance -= w;
11.   }
12
13.   public double getBalance() {
14.      return balance ;
15.   }
16.
17.   public static void main(String[] args) {
18.      Account checking = new Account() ;
19.      double cash = 1024.48 ;
20.
21.      checking.deposit(cash) ;
22.      System.out.println("balance=" + checking.getBalance()) ;
23.      checking.withdraw(cash) ;
24.      System.out.println("balance=" + checking.getBalance()) ;
25.   }
26.}
```

እንደገና ማሳሰቢያ፦ ተራ ቁጥሩ የተሰጠውን ፕሮግራም ለማብራራት ይረዳ ዘንድ እንጂ ፕሮግራሙ ውስጥ መጨመር ስላለበት አይደለም።

አሁን ፥ ደረጃ በደረጃ ፥ ይኸን ፕሮግራም ለማብራራት እንሞክራለን። አንባቢው ለጃቫ እንግዳ ከሆነ ፥ እያንዳንዱን የፕሮግራም ክፍል የግድ አሁኑን መረዳት አይኖርበትም። እንዲያውም የዚህ ምሳሌ ዓላም የጃቫ ፕሮግራም እንዴት እንደሚመደብ ማሳየት ስለሆነ ትኩረቱ ወደዛ መሆኑ ይረዳል። የፕሮግራሙ ዝርዝር ማብራሪያ እነሆ፦

- [መሥመር 1]፦ የጃቫ መደብ ለመፍጠር የምንጠቀመው ያዋጅ ቃል `class` ነው። ተራ ቁጥር 1 እንደሚያሳየው ፥ የ`class` ቃል `Account` ተብሎ የሚጠራ መደበ ይፈጥራል ወይም ይመድባል። የመደቡ ሰውነት **በግራ የጥልፍ ቅንፍ** ({) ተከፍቶ **በቀኝ የጥልፍ ቅንፍ** (}) ይዘጋል። እነዚህ የመደቡ አጥር ናቸው። የመደቡን ሰውነት ለጊዜው ብናወጣና በነጠብጣብ ብንተካው ፥ መደቡ ይኸን ይመስላል።

  ```
  class Account {...}
  ```

  የመደቡ አብይ መዋቅር ይህ ነው። የመደቡ ሥራና ችሎታ ሰውነቱ ውስጥ ባሉት አባላት ይወሰናል። ቀጥሎ እነሱን በዝርዝር እንመለከታለን።

- [መሥመር 2]፦ የመደቡን ዴታ መጠበቂያ ያውጃል። ዴታ መጠበቂያዎችን ተውላጠ-ቃላት (variables) ብለን እንጠራቸዋለን። አንዳንድ ጊዜ «ተለዋጭ» ወይም «ተለዋዋጭ» ብለው ይጠሯቸዋል። መደቦች በየጊዜው ያስበትን አቋም የሚጠብቁበትና የሚያሳዩበት ክፍል ይህ ነው። የዴታ ማጠራቀሚያቸው ቁጥርና አይነት እንዳስፈላጊነቱ ይለያያል። እንደዚህ አይነት ክፍሎች ፥ በእንግሊዘኛ `variable` (ቫርየብል) ናቸው። ከዚህ መደብ እንዳር ፥ «balance» የሚለው ዴታ ማጠራቀሚያ የተደለደለውን ዋጋ ይጠብቃል።
- [መሥመር 4-6]፦ ይህ ክፍል ፋንክሽን (function) ተብሎ ይጠራል። ተልዕኮው ፥ በተጠራ ቊጥር በሰውነቱ ያሉትን የመመሪያ ቃላት በተግባር ማዋል ነው። ማለትም የ«balance»ን ዕሴት በተላከው አዲስ ዋጋ ይጨምራል።
- [መሥመር 8-11]፦ ይህ ክፍል ፋንክሽን ነው። በተጠራ ጊዜ ከ«balance» ተውላጠ-ቃል ላይ የተላከውን ዕሴት ይቀንሳል።
- [መሥመር 13-15]፦ ይህ ክፍል ፋንክሽን ነው። በተጠራ ጊዜ የ«balance»ን ዕሴት ይመልሳል። በሌላ አነጋገር `balance` ውስጥ ያለውን ዕሴት እንድናነብ ያደርጋል።
- [መሥመር 17-25]፦ እንደሌሎቹ ሁሉ ይህ ክፍል ፋንክሽን ነው። ፕሮግራሙ ሥራውን የሚጀምረው ከዚህ ፋንክሽን ተነስቶ ከመሆኑም በላይ ልዩ ልዩ ተግባራትን ይፈጽማል። መደቡን አራብቶ በተግባር ላይ ያውላል።

የጃቫ መደብ አፈጣጠር ወይም መደብ አመዳደብ በትንሹ ይኸን ይመስላል። መደቡ መጀመሪያና መጨረሻ አለው። የመደቡ አባላት ከዚህ ምሳሌ አንጻር ዴታ መጠበቂያና ተግባራት መፈጸሚያ ፋንክሽኖች ናቸው። ከፕሮግራሙ ጋር በቅርብ ለመተዋወቅ የግንባታውን ሥራ እናጠናቅና በተግባር ላይ እናውለው። የግንባታውን ሂደት ከዚህ በፊት ባለው ክፍል ውስጥ ተሰጥቷል። እዚህ የምንደግመው ላመቺነት ስንል ነው።

- ከዊንዶስ «Windows Explorer» እናስነሳ።
- ከቋሚ ዲስካችን ፥ የ«eclipse»ን ማኅደር (folder) እንክፈት።
- በማኅደሩ ውስጥ ያሉት ፋይሎችና ንኡስ-ማኅደሮች ይዘረዘራሉ። ስሙ «eclipse.exe» የሚለውን ፋይል እናስነሳ። ኢክልፕስ ለሥራ ይሰማራል።
- በኢክልፕስ ያሠራር ዘይቤ መሠረት ፥ በመጀመሪያ የጃቫ ፕሮጀክት (project) እንፈጥራለን። ፕሮጀክት ፥ ተዛማጅ

ፕሮግራሞችን ባንድ ወገን አደራጅተን በቀላሉ የምንገነባበት ዘዴ ነው። በኢክሊፕስ ሥር ፥ የጃቫ ፕሮጀክት አፈጣጠር እንዲህ ነው። አንባቢው የፈጠረው ፕሮጀክት ካለ ፥ ሌላ አዲስ መፍጠር አላስፈላጊ ስለሆነ የፕሮጀክት ፈጠራውን ሂደት ማለፍ ይበጃል።

1. ከሜኒዩባር የFileን ሜኒዩ እንመርጣለን። ከሚዘረጋው ሜኒዩ ደግሞ New ፥ ከዛ Project... እንወስዳለን።
2. መለስተኛ መስኮት ይከፈታል። በስተግራ በኩል ያለው ክፍል ውስጥ የተመረጠው Java መሆኑን አረጋግጠን Next ያለበትን ቁልፍ እንጫን።
3. ሌላ መለስተኛ መስኮት ይከፈታል። በ«ፕሮጀክት ሥም» ሥር keffa ብለን እንጻፍ። አሁን የፕሮጀክታችንን ሥም «keffa» ሆነ ማለት ነው። ካሉት ቁልፎች መካከል አንዱ Finish ይላል። እሱን ጠቅ እናድርግ። ከዚህ በኃላ የሚመጣውን ምርጫ በመቀበል የፕሮጀክት ፈጠራውን ሂደት እናጠናቅቃለን።

- ቀጥለን አዲስ መደብ (class) እንፈጥራለን። ከሜኒዩባር File ፥ ከዛ New ፥ ቀጥሎ Class እንወስዳለን። መለስተኛ መስኮት ይከፈትና ለመደባችሁ ሥም አውጡ ይላል። የመደባችን ሥም Account ብለን እንጻፍና Finish የሚለውን ቁልፍ እንጫን። አሁን ንባባዊ-ኮድ መጻፊያ ገጽ ይከፈታል። አንባቢው እንደሚያየው ፥ ኢክሊፕስ የተወሰነ ነገር በራሱ ገጹ ላይ ያሠፍራል። ገጹ ላይ የሚታዩትን ነገሮች በሙሉ እናጥፋና ከላይ የተሰጠውን ኮድ እንዳለ እንጻፍ።
- ንባባዊ-ፕሮግራሙን በትክክል ከጻፍን በኃላ ፥ ወደ ዲስካችን እንገልብጥ። ሁለት መንገዶች አሉ። 1ኛ) ከቱልባሩ ላይ Save የሚለውን ቁልፍ ጠቅ ማድረግ ወይም ከሜኒዩባር File እንምረጥና ከሚዘረጋው Saveን እንውሰድ። አሁን ፥ ኢክሊፕስ ፋይሉን ወደ ዲስክ ይገለብጣል። አስከትሎ ፕሮግራማችንን ኮምፓይል ያደርጋል። አንዱ የኢክሊፕስ ልማድ ፥ ፕሮግራሞችን ወደ ዲስክ ገልብጥ ባልነው ጊዜ ፥ ከገለበጠ በኃላ ኮምፓይል ያደርጋቸዋል። ይኽን ልማድ የማንፈለግ ከሆነ ፥ ኢክሊፕስን መቃኘት እንችላለን።
- ኢክሊፕስ ፕሮግራማችንን ኮምፓይል ሲያደረግ ፥ ማለትም ፕሮግራማችን የጃቫን ሕግና ደንብ ማክበርና አለማክበሩን ሲመለከት ስህተት ካገኘ ፥ የኮምፓይል ሥራውን አቋርጦ ስህተቱን ያመለክታል። የኮምፓይል ሥራው በሥነ ሥርዓት ይጠናቀቅ ዘንድ ስህተቶቹ በሙሉ መታረም አለባቸው። አብዛኛውን ጊዜ የስህተት ምንጮች ከሚሆኑት ችግሮች መካከል አንዱ የፊደላት ግድፈት ነው። በተለይ በላቲን ፊደል ላንድ ድምጽ ሁለት አይነት ፊደላት አሉ። ለምሳሌ ትልቁ «A» እና ትንሹ «a» እና የመሳሰሉት። ትንሹንና ትልቁን የላቲን ፊደላት ፥ ጃቫ ባንድ አይን አይመለከታቸውም። ስለዚህ «Account» የሚለውን ቃል «account» ብለን ከጻፍን ለጃቫ ሁለቱ የተለያዩ ቃላት ናቸው። መጠንቀቅ አለብን።
- ፕሮግራማችን ውስጥ ስህተት ካልተገኘ ፥ ኮምፓይል ተደረጓል ማለት ነው። የጃቫ ኮምፓይለር አንድን ፕሮግራም መርምሮ ከጨረሰ በኃላ ፥ የሱ ዝርያ (አቻ) የሆነ የጃቫ «ባይትኮድ» (bytecode) ወይም የማሽን ቋንቋ ያመነጫል። ሥሙም «.class» ተቀጥያ ይኖረዋል። ለምሳሌ የኛ ፕሮግራም ሥም «Account.java» ነበር ፤ ባኋያው የሚመነጨው የባይትኮድ ፋይል «Account.class» ይሆናል።
- በመጨረሻ ፕሮግራማችን ለመፈተን ዝግጁ ነን። ከሜኒዩባር Run እንምረጥ ፥ ከዛ Run As ፥ ቀጥሎ Java Application እንምታ። ፕሮግራማችን ወደ ሥራ ይሠማራል። ውጤቱን ታች ያለው መስኮት ውስጥ ያሳያል። በመሆኑም፦

```
balance=1024.48
balance=0.0
```

2.3 የመደብ አባላት

የጃቫ የመደብ አባላት የምንላቸው የሚከተሉት ክፍሎች ናቸው። አንድ መደብ ዚሮ ወይም ከዚሮ በላይ እነዚህ አባላት ሊኖሩት ይችላል። የመደቡን አባላት አሥፈላጊነትና ቁጥር የሚወስነው ከሞላ ጎደል ችግሮችን ለመፍታት የሚሰጠው መፍትሄ ነው። ለመደብ መኖር ሁሉም አባላት የግድ መገኘት የለባቸውም።

1. ተውላጠ-ቃላት፦ በመደብ ደረጃ ዴታዎች ለመጠበቅ እንዲሁም ለመገልገል ያሥችላሉ። በሌላ አባባል ፥ የመደቡን ባህሪያ ያንፀባርቃሉ ወይም የመደቡን ይዞታ በማንኛውም ሂደት ውስጥ ይገልጻሉ። ማንኛውም መደብ ዴታ በግሉ መጠበቅ ከፈለገ የራሱን ተውላጠ-ቃላት መፍጠር ግዴታው ነው። የተውላጠ-ቃላቱ ማንነት በሚፈጠሩብት የዴታ ዓይነት ይወሰናል።
2. ፋንክሽኖች፦ የመደቡን ተግባራት ይፈጽማሉ። የመደቡን ተውላጠ-ቃላት ዕሴት ይደለድላሉ ፤ ያነባሉ። ባጠቃላይ የተሰጣቸውን መመሪያ በቅደም-ተከተል በሥራ ያውላሉ። መደቡን የሚያንቀሳቅሱትና ሕይወት የሚሰጡት እነሱ ናቸው።
3. ውስጣዊ መደባት፦ እንዳስፈላጊነቱ አንድ መደብ ውስጥ መደባት ሊኖሩት ይችላል። ይኽን ነጥብ ወደፊት በሰፊው እንመጣበታለን።
4. ስታቲክ ኢንሻላይዘር (static initialization)።

2.4 የጃቫ ፕሮግራም ሥራ መጀመሪያ ፋንክሽን

የጃቫ ፕሮግራም ለሥራ ስናስነሳ ፥ ተግባሩን የሚጀምረው ከmain() ፋንክሽን ነው። ስለሆነም ፥ ማንኛውም የጃቫ መደብ በራሱ ችሎታ ለሥራ መሰመራት የሚሻ ከሆነ ፥ የግድ የmain() ፋንክሽን ሊኖረው ይገባል።

ይህ ማለት ፥ ያንድ መደብ አባላት ፥ ለምሳሌ ፈንክሽኖች ፥ እንዲሁም ውስጣዊ-መደቦች ተራ አቀማመጥ የመመሪያዎችን የሥራ አፈጻጸም ቅደም ተከተል አይገዛም። ለዚህ ዋናው ምክንያት ፥ የጃቫ ፋንክሽኖች ወደሥራ የሚሰማሩት ሲጦሩ ብቻ ስለሆነ የተራ አቀማመጣቸው ለሥራ አፈጻጸም ቅደም-ተከተል ቦታ የለውም። በመሆኑም ፥ አንዳንድ የፕሮግራም ጸሐፊዎች የmain() ፋንክሽንን ከሁሉ ፋንክሽኖች አስቀድመው ተራ ሲሰጡ ፥ አብዛኛዎች ግን ከሁሉም ፋንክሽኖች መጨራሻ ተራ ይሰጡታል።

ባጭሩ የጃቫን ፕሮግራም ሥራውን ሁሉ ጊዜ የሚጀምረውና የሚጨርሰው ከmain() ፋንክሽን ነው። የመመሪያዎች ሥራ አፈጻጸም ቅደም-ተከተል ግን በፋንክሽኖች የጥሪ ተረታ ይወሰናል።

◀ ተመለሥ ማውጫ ቀጥል ▶

contact@senamirmir.com
Copyright © 2002-2004 Senamirmir Project