

ምሳሌ

የጃቫ ፕሮግራም መዋቅርና አቋቋም

የጃቫ ፕሮግራም የማያሻማና 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 ምርምር የተዋቀሩ የመደቦች ስብስብ ማለታችን ነው።

2.1 ማብራሪያ (Comments)

ፕሮግራሞች ለተለያዩ ችግሮች መፍትሄ የሚሰጡ ሥራዎች ናቸው። የመሠረታዊ ቁጥር እየጨመረ ሲመጣ ፣ አነስጥር በቅጡ አንብቦ የመረዳቱ ችግር እየበረታ ይመጣል። ስለዚህ የፕሮግራሞችን ተነባቢነት ለመጨመር ከሚደረጉት ዋረቶች መካከል አንዱ ማብራሪያዎችን ፕሮግራሞቹ ውስጥ መክተት ነው። ለዚህ ፣ ጃቫ ሁለት አይነት ማብራሪያ መጻፊያ መንገዶች አሉት።

1. በሁለት «የቋረት» (//) ምልክቶች የሚጀምሩ መሥመሮች በውስጥ ማብራሪያ ናቸው። ማብራሪያው ካንድ መሥመር በላይ ከሆነ ፣ እያንዳንዱ መሥመር የግድ በሁለት የቋረት (//) ምልክቶች መጀመር አለበት። እንደዚህ ያሉትን ምልክቶች የ«ማብራሪያ ምልክቶች» ብለን ልንጠራቸው እንችላለን።
2. ሁለተኛው መንገድ ካንድ መሥመር በላይ የሆነ አስተያየት ለመጻፍ አመቺ መንገድ ነው። ማብራሪያው በ«/*» ተከፋቶ በ«*/» ይዘጋል።

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

ማብራሪያ አጻጻፍ
<pre> 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.out.println(i + ". " + args[i]) ; 12. } 13. }</pre>

ይህ ምሳሌ ፥ ሁለቱንም የማብራሪያ አጻጻፍ መንገዶች ያሳየናል። ከተራ-ቁጥር 2 እስከ 5 ያለው አንደኛው አይነት ማብራሪያ አጻጻፍ ሲሆን ቁጥር 9 ላይ ያለው ደግሞ ሌላኛው ነው። እነዚህ ማብራሪያዎች ለፕሮግራም ሎጂክ የሚሰጡት ምንም አስታዋቂ 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 ፥ ንድፋን ካስቀመጠ በኋላ ፥ 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 ዓለም 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 መደብ ይፈጥራል ወይም ይመድባል። የመደቡ ሰውነት በግራ የጥልፍ ቅንፍ (`{}`) ተክፍቶ 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። ቁጥሉ እነሱን በዝርዝር እንመለከታለን።

- [መሥመር 2]፦ የመደቡን ዴታ መጠበቂያ ያውጃል። ዴታ መጠበቂያዎችን ተውላጠ-ቃላት (variables) ብለን እንጠራቸዋለን። እንዳንድ ጊዜ «ተለዋጭ» ወይም «ተለዋዋጭ» ብለው ይጠሯቸዋል። መደቦች በየጊዜው ያሉበትን አቋም የሚጠብቁበትና የሚያሳዩበት ክፍል ይህ ነው። የዴታ ማጠራቀሚያቸው ቁጥርና አይነት እንዳስፈላጊነቱ ይለያያል። እንደዚህ አይነት ክፍሎች ፣ በእንግሊዘኛ variable (ቫርያብል) ናቸው። ከዚህ መደብ እንጻር ፣ «balance» የሚለው ዴታ ማጠራቀሚያ የተደለደለውን ዋጋ ይጠብቃል።
- [መሥመር 4-6]፦ ይህ ክፍል ፋንክሽን (function) ተብሎ ይጠራል። ተልዕኮው ፣ በተጠራ ቁጥር በሰውነቱ ያሉትን የመመሪያ ቃላት በተግባር ማዋል ነው። ማለትም የ«balance»ን ዕሴት በተለከው አዲስ ዋጋ ይጨምራል።
- [መሥመር 8-11]፦ ይህ ክፍል ፋንክሽን ነው። በተጠራ ጊዜ ከ«balance» ተውላጠ-ቃል ላይ የተለከውን ዕሴት ይቀንሳል።
- [መሥመር 13-15]፦ ይህ ክፍል ፋንክሽን ነው። በተጠራ ጊዜ የ«balance»ን ዕሴት ይመልሳል። በሌላ አነጋገር balance ውስጥ ያለውን ዕሴት እንድናገኝ 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 እናጠናና በተግባር ላይ እናውለው። የግንባታውን ሂደት ከዚህ በፊት ባለው ክፍል ውስጥ ተሰጥቷል። እዚህ የምንደግመው ላመቺነት ስንል ነው።

- ከዊንዶሽ «Windows Explorer» እናስነሳ።
- ከቋሚ ዲስኮች ፣ የ«eclipse»ን ማሳደር (folder) እንክፈት።
- በማሳደሩ ውስጥ ያሉት ፋይሎችና ንጉስ-ማሳደሮች ይዘረዘራሉ። ስሙ «eclipse.exe» የሚለውን ፋይል እናስነሳ። አክልፕሽ ለሥራ ይሰማራል።
- በአክልፕሽ ያሠራር ዘይቤ መሠረት ፣ በመጀመሪያ የጃቫ ፕሮጀክት (project) እንፈጥራለን። ፕሮጀክት ፣ ተዛማጅ

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

1. ከሜኒዩር 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 ቀጥሎ እንጫን።
3. ሌላ መለስተኛ መስኮት ይከፈታል። በ«ፕሮጀክት ስም» ሥር keffa ብለን እንጻፍ። አሁን የፕሮጀክታችንን ስም «keffa» ሆነ ማለት ነው። ካሉት ቀጥሎ መካከል አንዱ Finish ይላል። እሱን ጠቅ እናድርግ። ከዚህ በኋላ የሚመጣውን ምርጫ በመቀበል የፕሮጀክት ፈጠራውን ሂደት እናጠናቅቃለን።

- ቀጥለን አዲስ መደብ (class) እንፈጥራለን። ከሜኒዩር File ፣ ከዛ New ፣ ቀጥሎ Class እንወስዳለን። መለስተኛ መስኮት ይከፈትና ለመደባችሁ ስም አውጡ ይላል። የመደባችን ስም Account ብለን እንጻፍና Finish የሚለውን ቀጥሎ እንጫን። አሁን ንባባዊ-ኮድ መጻፊያ ገጽ ይከፈታል። አንባቢው እንደሚያየው ፣ ኢክሌፕሽን የተወሰነ ነገር በራሱ ገጹ ላይ ያሠፍራል። ገጹ ላይ የሚታዩትን ነገሮች በመሉሉ እናጥፋና ከላይ የተሰጠውን ኮድ እንዳለ እንጻፍ።
- ንባባዊ-ፕሮግራሙን በትክክል ከጻፍን በኋላ ፣ ወደ ዲስካችን እንገልብጥ። ሁለት መንገዶች አሉ። 1ኛ) ከቱልባሩ ላይ Save የሚለውን ቀጥሎ ጠቅ ማድረግ ወይም ከሜኒዩር File እንምረጥና ከሚዘረጋው Saveን እንወስድ። አሁን ፣ ኢክሌፕሽን ፋይሉን ወደ ዲስክ ይገለብጣል። አስከትሎ ፕሮግራሞችን ከምጋይል ያደርጋል። አንዱ የኢክሌፕሽን ልማድ ፣ ፕሮግራሞችን ወደ ዲስክ ገልብጦ ባልነው ጊዜ ፣ ከገለበጠ በኋላ ከምጋይል ያደርጋቸዋል። ይኸን ልማድ የማንፈለግ ከሆነ ፣ ኢክሌፕሽን መቃኘት እንችላለን።
- ኢክሌፕሽን ፕሮግራሞችን ከምጋይል ሲያደረግ ፣ ማለትም ፕሮግራሞችን የጃቫን ሕግና ደንብ ማክበርና አለማክበሩን ሲመለከት ስህተት ካገኘ ፣ የኮምፓይል ስራውን አቋርጦ ስህተቱን ያመለክታል። የኮምፓይል ስራው በሥነ ሥርዓት ይጠናቀቅ ዘንድ ስህተቶቹ በመሉሉ መታረም አለባቸው። አብዛኛውን ጊዜ የስህተት ምንጮች ከሚሆኑት ችግሮች መካከል አንዱ የፊይልት ግድፈት ነው። በተለይ በላቲን ፊይልት ላይ ድምጽ ሁለት አይነት ፊይልት አሉ። ለምሳሌ ትልቁ «A» እና ትንሹ «a» እና የመሳሰሉት። ትንሹንና ትልቁን የላቲን ፊይልት ፣ ጃቫ ባንድ አይነት አይመለከታቸውም። ስለዚህ «Account» የሚለውን ቃል «account» ብለን ከጻፍን ለጃቫ ሁለቱ የተለያዩ ቃላት ናቸው። መጠንቀቅ አለብን።
- ፕሮግራሞችን ውስጥ ስህተት ካልተገኘ ፣ ኮምፓይል ተደረጓል ማለት ነው። የጃቫ ኮምፓይሎር አንድን ፕሮግራም መርምሮ ከጨረሰ በኋላ ፣ የሱ ዝርያ (አቻ) የሆነ የጃቫ «ባይትኮድ» (bytecode) ወይም የማሽን ቋንቋ ያመነጫል። ሥሙም «.class» ተቀጥሎ ይኖረዋል። ለምሳሌ የኛ ፕሮግራም ስም «Account.java» ነበር ፤ ባኳያው የሚመነጨው የባይትኮድ ፋይል «Account.class» ይሆናል።
- በመጨረሻ ፕሮግራሞችን ለመፈተን ዝግጁ ነን። ከሜኒዩር Run እንምረጥ ፣ ከዛ Run As ፣ ቀጥሎ Java Application እንምታ። ፕሮግራሞችን 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። እንደ መደብ ዘሮ ወይም ከዚሮ በላይ እነዚህ አባላት ሊኖሩት ይችላል። የመደቡን አባላት አሥፈላጊነትና ቀዳሚ የሚወስነው ከሞላ ጎደል ችግሮችን ለመፍታት የሚሰጠው መፍትሄ ነው። ለመደብ መኖር ሁሉም አባላት የግድ መገኘት የለባቸውም።

1. ተውላጠ-ቃላት፡- በመደብ ደረጃ ይታዩ ለመጠበቅ እንዲሁም ለመገልገል ያሥችላሉ። በሌላ አባባል ፣ የመደቡን ባህሪያ ያንጸባርቃሉ ወይም የመደቡን ይዘታ በማንኛውም ሂደት ውስጥ ይገልጻሉ። ማንኛውም መደብ ይታ በግሉ መጠበቅ ከፈለገ የራሱን ተውላጠ-ቃላት መፍጠር ግዴታው ነው። የተውላጠ-ቃላቱ ማንነት በሚፈጠሩበት የደታ 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 እንደ መደብ ውስጥ መደባት ሊኖሩት ይችላል። ይኸን 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 የሚችል 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 ሲጠሩ ብቻ ስለሆነ የተራ አቀማመጣቸው ለሥራ አፈጻጸም ቅደም-ተከተል ቦታ የለውም። በመሆኑም ፣ አንዳንድ የፕሮግራም ጸሐፊዎች የmain() ፋንክሽንን ከሁሉ ፋንክሽኖች አስቀድመው ተራ ሲሰጡ ፣ አብዛኛዎቹ ግን ከሁሉም ፋንክሽኖች መጨረሻ ተራ ይሰጡታል።

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



contact@senamirmir.com

Copyright © 2002-2004 Senamirmir Project