

# 1 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 ሥርዓት የተዋቀሩ የመደቦች ስብስብ ማለታችን ነው።

# 2 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።

- በሁለት የቁረት (//) ምልክቶች የሚጀምሩ መሥመሮች በሙሉ ማብራሪያ ናቸው። ማብራሪያው ካንድ መሥመር በላይ ከሆነ ፣ እያንዳንዱ መሥመር የግድ በሁለት የቁረት (//) ምልክቶች መጀመር አለበት። እንደነዚህ ያሉትን ምልክቶች የማብራሪያ ምልክቶች ብለን ልንጠራቸው እንችላለን።
- ሁለተኛው መንገድ ካንድ መሥመር በላይ የሆነ አስተያየት ለመጻፍ አመቺ መንገድ ነው። ማብራሪያው በ«/\*» ተከፍቶ በ«\*/» ይዘጋል።

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

```

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. }

```

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

### 3 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 በንጽቦት 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 መሻጣት አይችልም። የራሱን አጥር በሥነ ሥርዓት የማጠር ኃላፊነትም አለበት።

የጃቫ አፈጣጠር ወይም አመዳደብ እንደሚከተለው ነው። ይኸን ኮድ ለማብራራት ይረዳ ዘንድ እያንዳንዱ መሥመር መጀመሪያ ላይ ተራርቁጥር ተጨምሯል። የተራ ቁጥሩ ዓላማ ይህ ይህ ብቻ ስለሆነ ፣ አውነተኛ ፕሮግራሞቻችን ውስጥ በፍጹም መጨመር የለብንም ፣ ብንሻም መጨመር አይፈቀድልንም።

```
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 የገድ አሁኑን መረዳት አይኖርበትም። እንዲያውም የዚህ ምሳሌ ዓለም የጃቫ ፕሮግራም እንዴት እንደሚመደብ ማሳየት ስለሆነ ትኩረቱ ወደዛ መሆኑ ይረዳል። የፕሮግራሙ ዝርዝር ማብራሪያ እነሆ፡-

1 ፡ የጃቫ መደብ ለመፍጠር የምንጠቀመው ያዋጅ ቃል class ነው። ተራ ቁጥር 1 እንደሚያሳየው ፣

የclass ቃል Account ተብሎ የሚጠራ መደብ ይፈጥራል ወይም ይመድባል። የመደቡ ሰውነት በግራ የጥልፍ ቅንፍ ( ) ተክሎች 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።

- 2 ፦ የመደቡን ዴታ መጠበቂያ ያውጃል። ዴታ መጠበቂያዎችን ተውላጠጭቃላት (variables) ብለን እንጠራቸዋለን። አንዳንድ ጊዜ ተለዋጭ ወይም ተለዋጭ 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 አንጻር ፣ «balance» የሚለው ዴታ ማጠራቀሚያ የተደለደለውን ዋጋ ይጠብቃል።
- 4-6 ፦ ይህ ክፍል ፋንክሽን (function) ተብሎ ይጠራል። ተልዕኮው ፣ በተጠራ ቁጥር በሰውነቱ ያሉትን የመመሪያ ቃላት በተግባር ማዋል ነው። ማለትም የ«balance»ን ዕሴት በተላከው አዲስ ዋጋ ይጨምራል።
- 8-11 ፦ ይህ ክፍል ፋንክሽን ነው። በተጠራ ጊዜ ከ«balance» ተውላጠጭቃል ላይ የተላከውን ዕሴት ይቀንሳል።
- 13-15 ፦ ይህ ክፍል ፋንክሽን ነው። በተጠራ ጊዜ የ«balance»ን ዕሴት ይመልሳል። በሌላ አነጋገር balance ውስጥ ያለውን ዕሴት እንድናነብ ያደርጋል።
- 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።

- ከዊንዶሽ «Windows Explorer» እናስነሳ።
- ከቋሚ ዲስኮች ፣ የ«eclipse»ን ማኅደር ረቦቪኬቡብቲሩ እንክፈት።
- በማኅደሩ ውስጥ ያሉት ፋይሎችና ንኡስርማኅደሮች ይዘረዘራሉ። ስሙ «eclipse.exe» የሚለውን ፋይል እናስነሳ። ኢክስፕሮክሽን ለሥራ ይሰማራል።
- በኢክስፕሮክሽን ያሠራር ዘይቤ መሠረት ፣ በመጀመሪያ የጃቫ ፕሮጀክት (project) እንፈጥራለን። ፕሮጀክት ፣ ተዛማጅ ፕሮግራሞችን ባንድ ወገን አደራጅተን በቀላሉ የምንገነባበት ዘይ ነው። በኢክስፕሮክሽን 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 ብቷ ፣ ከ Project... እንወስዳለን።

2. መለስተኛ መስኮት ይከፈታል። በስተግራ በኩል ያለው ክፍል ውስጥ የተመረጠው Java መሆኑን አረጋግጠን Next ያለበትን ቀላል እንጫን።
3. ሌላ መለስተኛ መስኮት ይከፈታል። በፕሮጀክት ሥም ሥር keffa ብለን እንጻፍ። አሁን የፕሮጀክታችንን ሥም «keffa» ሆነ ማለት ነው። ካሉት ቀላሎች መካከል አንዱ Finish ይላል። እሱን ጠቅ እናድርግ። ከዚህ በኋላ የሚመጣውን ምርጫ በመቀበል የፕሮጀክት ፈጠራውን ሂደት እናጠናቅቃለን።

- ቀጥሎ አዲስ መደብ (class) እንፈጥራለን። ከሜኒዩር File ፣ ከዛ New ፣ ቀጥሎ Class እንወስዳለን። መለስተኛ መስኮት ይከፈትና ለመደባቸው ሥም አውጦ ይላል። የመደባችን ሥም Account ብለን እንጻፍና Finish የሚለውን ቀላል እንጫን። አሁን ንባባዊነቱን መጻፊያ ገጽ ይከፈታል። አንባቢው እንደሚያየው ፣ ኢክስፕሪት የተወሰነ ነገር በራሱ ገጹ ላይ ያሠፍራል። ገጹ ላይ የሚታዩትን ነገሮች በሙሉ እናጥፋና ከላይ የተሰጠውን ኮድ እንዳለ እንጻፍ።
- ንባባዊፕሮግራሙን በትክክል ከጻፍን በኋላ ፣ ወደ ዲስኮችን እንገልብጥ። ሁለት መንገዶች አሉ። ጀኛ ከተልባሩ ላይ Save የሚለውን ቀላል ጠቅ ማድረግ ወይም ከሜኒዩር File እንምረጥና ከሚዘረጋው Saveን እንውሰድ። አሁን ፣ ኢክስፕሪት ፋይሉን ወደ ዲስኮ ይገለብጣል። አስከትሎ ፕሮግራሞችን ኮምፓይል ያደርጋል። አንዱ የኢክስፕሪት ልማድ ፣ ፕሮግራሞችን ወደ ዲስኮ ገልብጥ ባልነው ጊዜ ፣ ከገለበጠ በኋላ ኮምፓይል ያደርጋቸዋል። ይኸን ልማድ የማንፈለግ ከሆነ ፣ ኢክስፕሪትን መቃኘት እንችላለን።
- ኢክስፕሪት ፕሮግራሞችንን ኮምፓይል ሲያደረግ ፣ ማለትም ፕሮግራሞችን የጃቫን ሕግና ደንብ ማክበር ና 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 የፊልት ግፊት ነው። በተለይ በላቲን ፊልት ላንድ ድምጽ ሁለት አይነት ፊልት አሉ። ለምሳሌ ትልቁ «A» እና ትንሹ «a» እና የመሳሰሉት። ትንሹንና ትልቁን የላቲን ፊልት ፣ 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 (ኦቻ) የሆነ የጃቫ ባይትኮድ (bytecode) ወይም የማሽን ቋንቋ ያመነጫል። ሥሙም «.class» ተቀጥሎ ይኖረዋል። ለምሳሌ የኛ ፕሮግራም ሥም «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
```

#### 4 የመደብ አባላት

የጃቫ የመደብ አባላት የምንለቸው የሚከተሉት ክፍሎች ናቸው። እንደ መደብ ዘሮ ወይም ከዘሮ በላይ እነዚህ አባላት ሊኖሩት ይችላል። የመደቡን አባላት አሥራላንትና ቀጥሮ የሚወሰነው ከሞላ ጉደል ችግሮችን ለመፍታት የሚሰጠው መፍትሄ ነው። ለመደብ መኖር ሁሉም አባላት የግድ መገኘት የለባቸውም።

- ተውላጠጭቃላት፦ በመደብ ደረጃ ዴታታዎች ለመጠበቅ እንዲሁም ለመገልገል ያሥቅላሉ። በሌላ አባባል ፣ የመደቡን ባህሪ ያንጸባርቃሉ ወይም የመደቡን ይዘት በማንኛውም ሂደት ውስጥ ይገልጻሉ። ማንኛውም መደብ ዴታ በግሉ መጠበቅ ከፈለገ የራሱን ተውላጠጭቃላት መፍጠር ግዴታው ነው። የተውላጠጭቃላቱ ማንነት በሚፈጠሩበት የዴታ ዓይነት ይወሰናል።
- ፋንክሽኖች፦ የመደቡን ተግባራት ይፈጽማሉ። የመደቡን ተውላጠጭቃላት ዕቤት ይደለድላሉ ፤ ያንባሉ። ባጠቃላይ የተሰጣቸውን መመሪያ በቅደምርተክተል በሥራ ያውላሉ። መደቡን የሚያንቀሳቅሱትና ሕይወት የሚሰጡት እነሱ ናቸው።
- ውስጣዊ መደባት፦ እንዳስፈላጊነቱ አንድ መደብ ውስጥ መደባት ሊኖሩት ይችላል። ይኸን ነጥብ ወደፊት በሰፊው እንመጣበታለን።
- ስታቲክ ኢንሻላይዘር (static initialization)።

## 5 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 ቅደምርተክተል 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 የጥሪ ተረታ ይወሰናል።