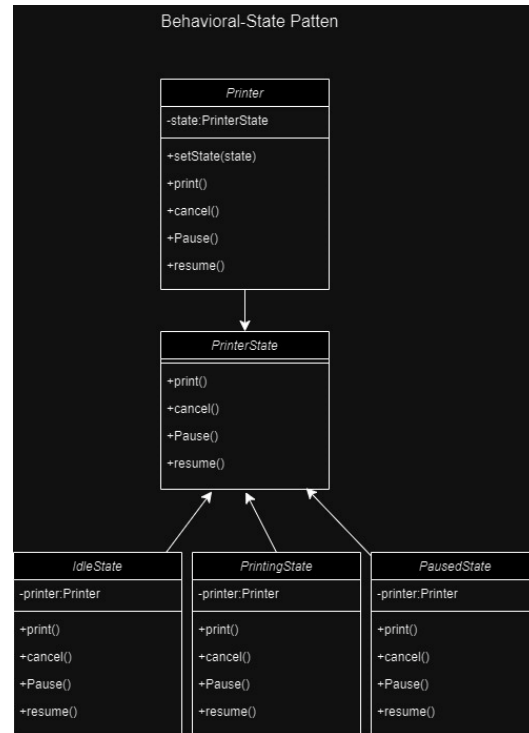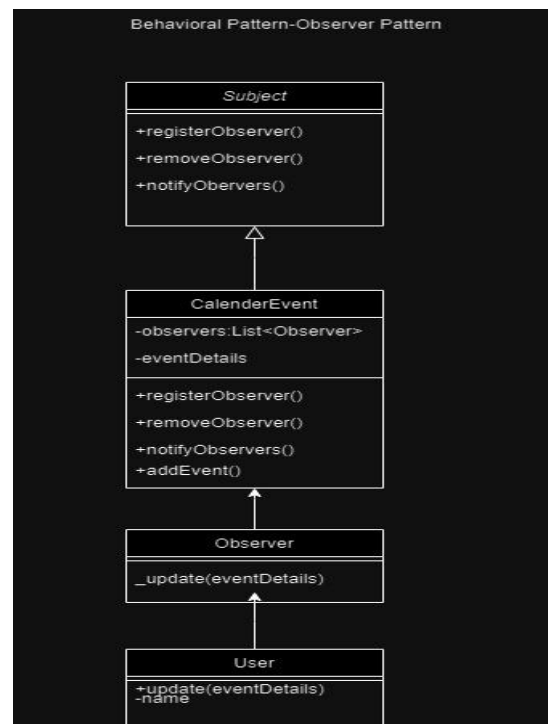# EX-1-Use Cases:

1. Underline Behavioral design Pattern:

   a) State Pattern-Printer State Maintenance

   b)Observer Pattern-CalenderEvent Management
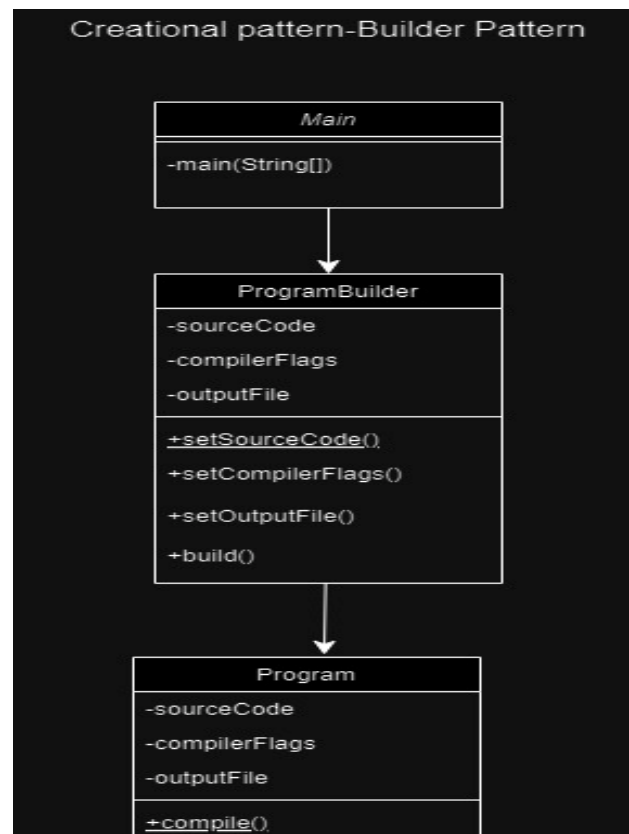
2. Creational design Pattern:
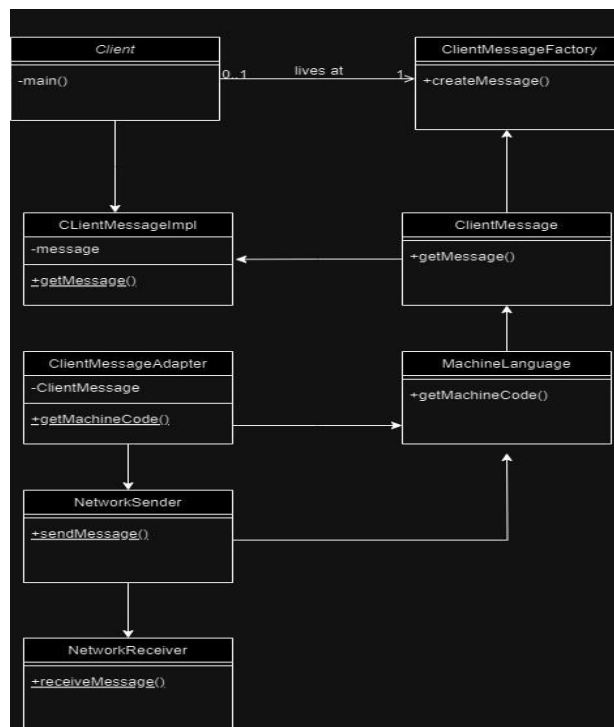   a) Singleton Pattern-PrinterSpool Job printing

Creational Pattern-Singleton Pattern

Main
-main()

printerSpool
-instance:PrinterSpool
-printJobs:List<PrintJob>
+getInstance()
+addPrintJob()
+printNextJob()
+getPrintJobCount()

PrintJob
-jobName
-documentName
+getJobName()
+PrintJob(jobName,documentName)
+getDocumentName()

b)Builder pattern-Program Compiling and Building

Creational pattern-Builder Pattern

Main
-main(String[])

ProgramBuilder
-sourceCode
-compilerFlags
-outputFile
+setSourceCode()
+setCompilerFlags()
+setOutputFile()
+build()

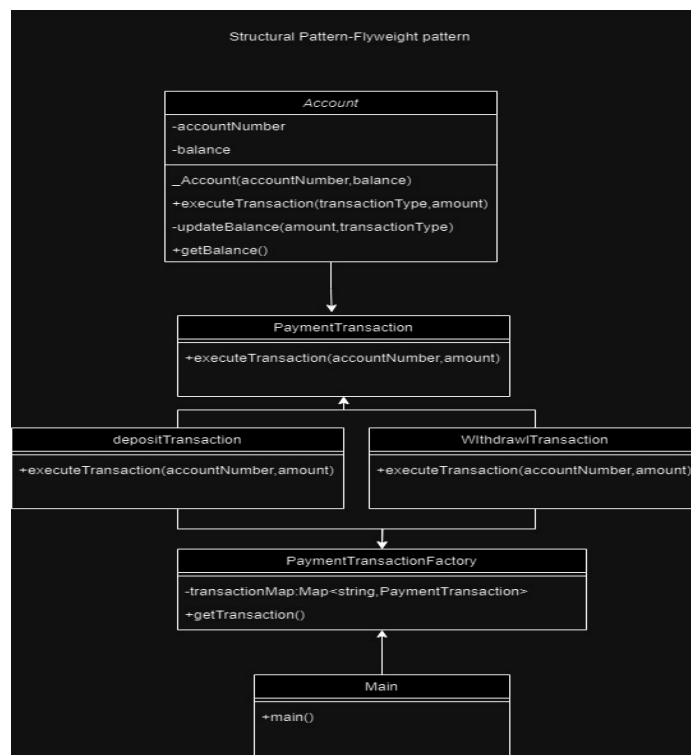| Program |
| --- |
| -sourceCode |
| -compilerFlags |
| -outputFile |
| +compile() |

3.Structural Pattern:

a)Adapter pattern-Message Transmission Over a network

b)Flyweight pattern-Transaction Management

[PS:The diagrams are drawn using Online draw.io UML tool]