1. SDLC [42]
2. Use the following table to compare various Software Development Life Cycle methodologies: [1x4x2 + 5x4x0.5 = 18]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Criteria | Waterfall | V-Model | Spiral | Iterative |
| What are the Various stages/phases | Requirements, Design, Implementation, Verification, Maintenance | Business Requirements (Acceptance Testing), System Requirements (System Testing), High-level Design (Integration Testing), Low-level Design (Functional testing), Coding (Unit testing) | Evaluate, Construct, Identify, Design | Requirements, Design & Development, Testing, Implementation |
| Which one allows Validation/verification for every stage? |  | x |  |  |
| Which one allows requirements changes? |  |  | x | x |
| Which one is best suited for static requirements? | x | x |  |  |
| Which one allows technology stack changes? |  |  | x | x |
| Which one allows variable iteration lengths? |  |  | x |  |

1. Use the following table to compare the key differences among SCRUM, KANBAN, and XP Agile flavors? [2x3x2 + 4x3x1=24]

|  |  |  |  |
| --- | --- | --- | --- |
| Criteria | SCRUM | KANBAN | XP |
| What are the various roles of individuals | Scrum Team – carries out design, coding, and testing Scrum Master – oversees scrum process during each sprint Product Owner – does initial planning and prioritization and communicates with rest of company | No formal defining roles | Development Team – similar to Scrum team  XP Coach – Similar to Scrum master  Product Owner – Same as Scrum product owner  Customer – constantly interacts with development team |
| What are the various Specific Meetings needed? | Sprint Planning, Daily Stand-up, Sprint Review, Sprint Retrospective | No formal meetings | Sprint Planning, Daily Stand-up, Sprint Review, Sprint Retrospective |
| Iteration Length/duration | 2-4 weeks | No ‘iterations’ – limits number of tasks in progress at any one time | 40 hours |
| Which Agile flavor on the right adopts advanced technologies, tools and processes ? |  |  | x |
| Which Agile flavor on the right needs highly skilled team members? |  |  | x |
| What are the Process measurement metrics (velocity chart or cycle time)? | Sprint burndown and velocity tracking | Cycle time | Sprint burndown and velocity tracking |

1. Architectural Patterns [48]
2. Use the following table to provide information about various Architectural Views. [Pts. 5x3x1=15]

|  |  |  |
| --- | --- | --- |
| Architectural View Name | One UML Diagram Name | One User Name |
| Logical View | Class Diagram | End users |
| Process View | Activity Diagram | Integrators |
| Deployment View | Deployment Diagram | System engineers |
| Implementation View | Component Diagram | Programmers |
| Use Case View | Use Case Diagram | All stakeholders |

1. Use the following table to compare Layered, Tiered, Pipe & Filter, and Client-Server for the following criteria: Sequential operation, parallel operation, Data flow directionality, and Resource/Infrastructure arrangement. [4x4x1=16]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Criteria | Layered | Tiered | Pipe & Filter | Client-Server |
| Sequential operation (Answer YES or NO in the columns on the right) | YES | YES | YES | YES |
| parallel operation (Answer YES or NO in the columns on the right) | NO | NO | YES | YES |
| Data flow directionality (Answer Unidirectional or Bidirectional in the columns on the right) | Bidirectional | Bidirectional | Unidirectional | Bidirectional |
| Resource/Infrastructure arrangement (Answer Single or Distributed in the columns on the right) | Single | Distributed | Single | Distributed |

1. What are the 3 mandatory items that you need to construct a Broker Architectural Pattern? What item will you add to make it an interoperable environment? What are the 2 items will you add to the design to add security to the Architecture? [3+1+2=6]

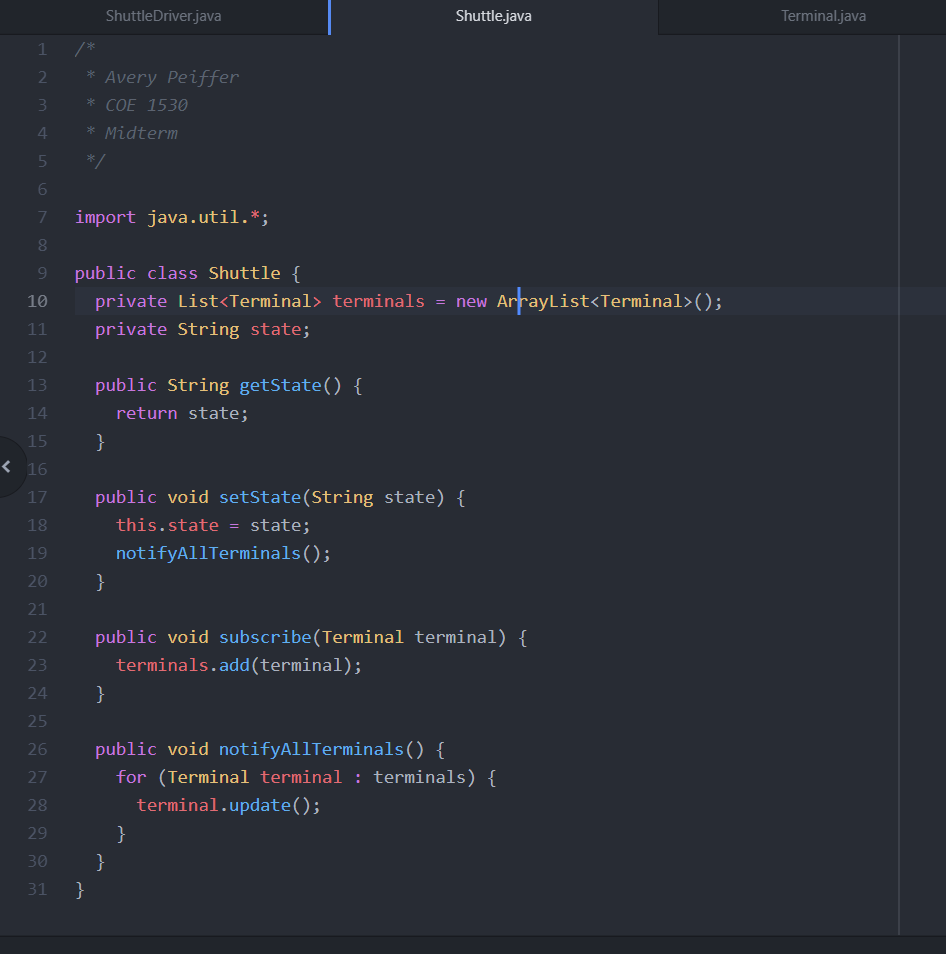
* Three mandatory items – client, server, broker
* Item for interoperable environment – bridge
* Two items to add security – server proxy, client proxy

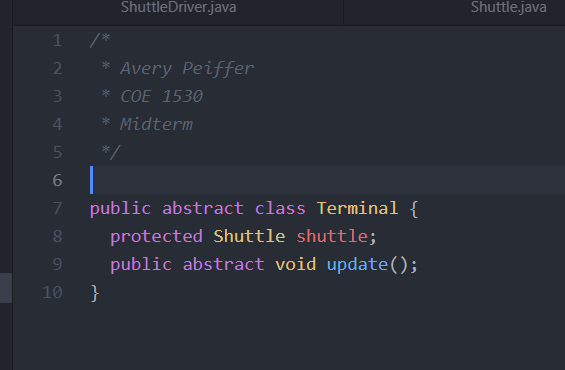
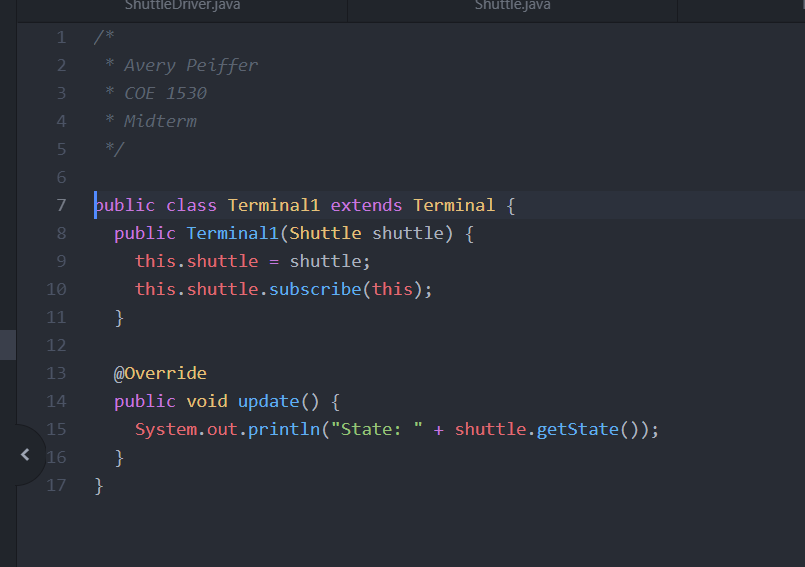
1. Which Architectural Pattern will you use for an automated 911 call service system that coordinates Emergency Medical Services, Fire Department, and Sheriff’s Department and why? What are the various components that you need in this architectural pattern? [2+5+4=11]

* The event-driven mediator pattern would be used because the 911 dispatcher acts as a central mediator that orchestrates multiple steps
* Components needed: event queues, event mediator, event channels, event processors

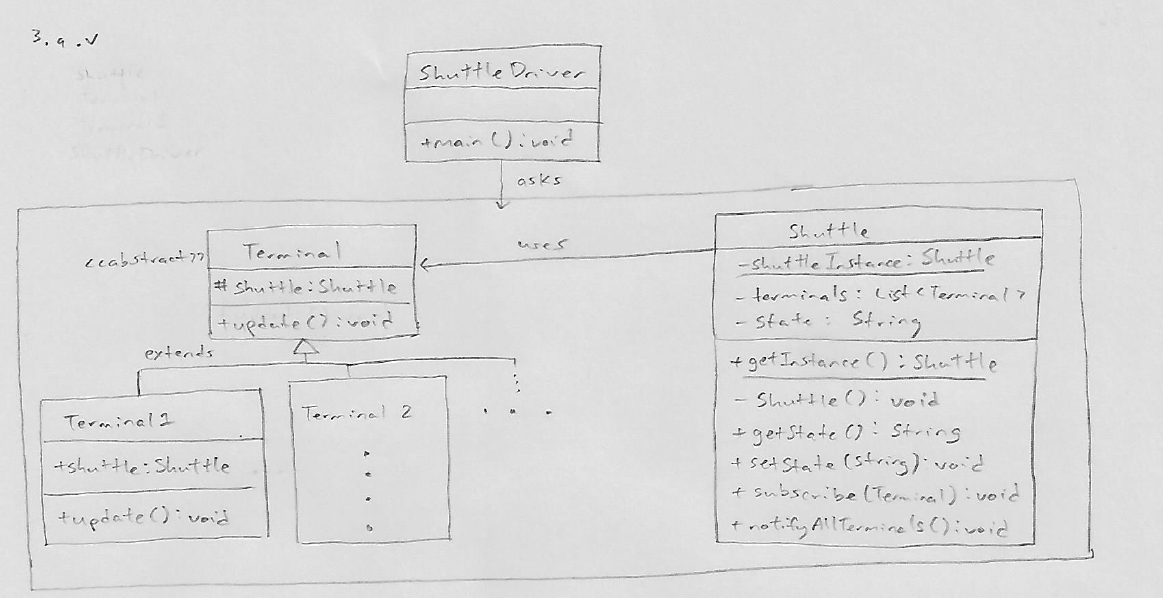
1. Design Patterns [40]
   1. An electric, automated and centrally controlled Airport Inter-Terminal Shuttle Service system has one Shuttle and 4 Terminals (Terminal1, Terminal2, Terminal3, and Terminal4). The system needs to be developed with the requirement that the Shuttle status needs to be relayed to all four terminals whenever there is status change (Moving, Stopped, Out-of-order, or Arrived at certain Terminal. The Shuttle and the four Terminals are represented by 5 Classes: Shuttle, Terminal1, Terminal2, Terminal3, and Terminal4. Provide answers to the following questions based on this case study: [35 pts]
      1. What Design Pattern will you adopt for this design and why? [What: 2 pts + Why: 3pts = 5]

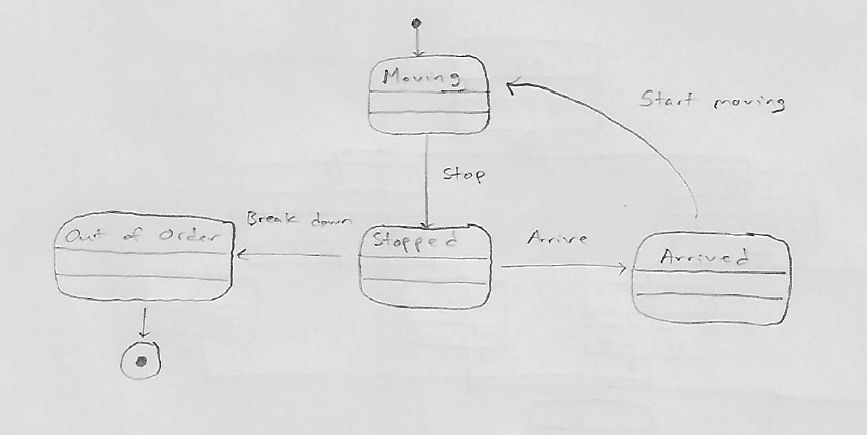
I will adopt the observer design pattern because the system is interested in the state of the shuttle. The shuttle’s state needs to be relayed to all four terminals when it is changed, so the observer pattern allows for there to be a one-to-many dependency between the shuttle and the terminals.

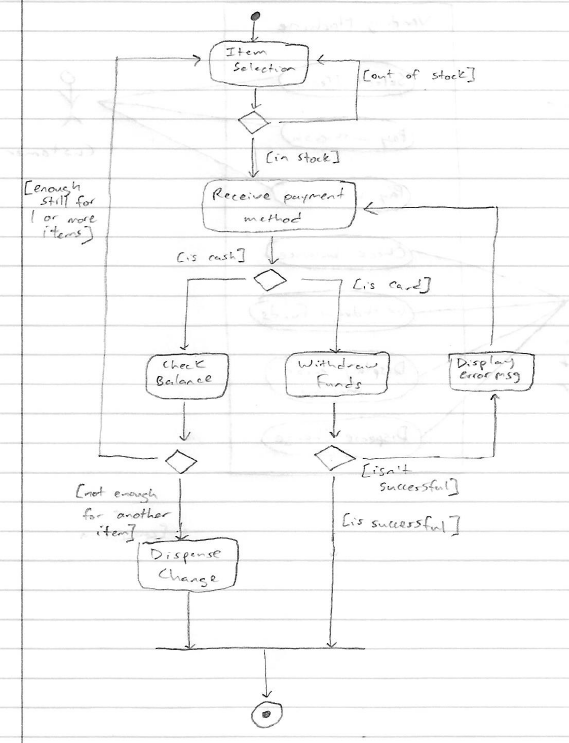
* + 1. Implement Shuttle Class in Java reflecting the chosen design pattern. [10]
    2. Implement Terminal1 Class in Java reflecting the chosen design pattern. [5]

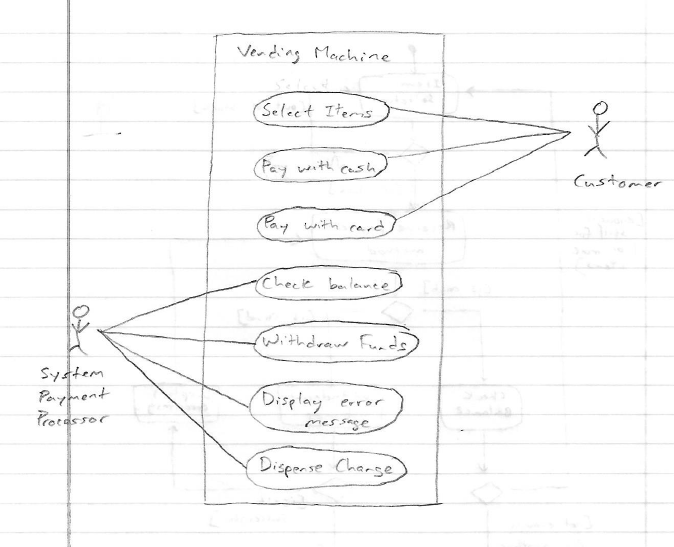


* + 1. Make the Shuttle class in 3.a.ii a Singleton class in Java. [5]
    2. Draw Class Diagram for your Solution. [5 pts]

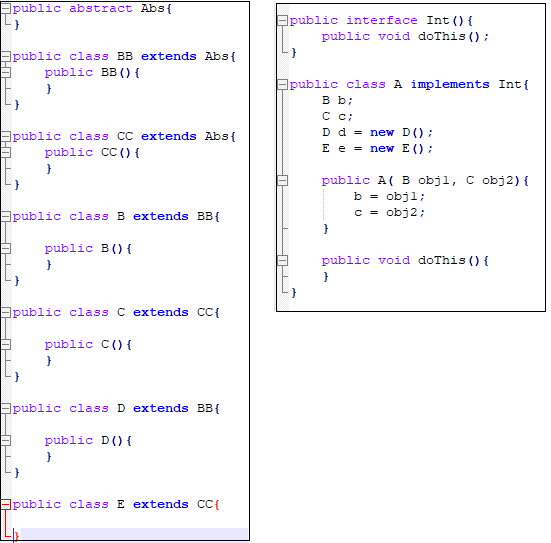


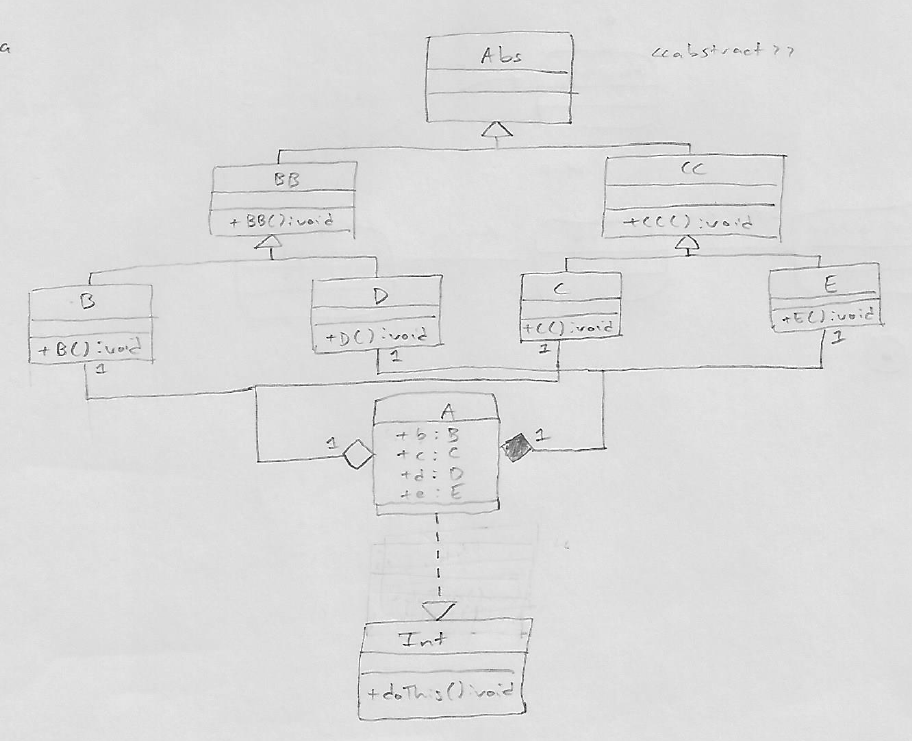
* + 1. Draw the State Diagram for your Solution. [5 pts]
  1. Draw activity and use case diagrams for a vending machine showing item selection, payment, balance and product dispensing. The vending machine dispenses various snacks and accepts both cash and credit cards. Show various scenarios of card processing: success and failure [15].



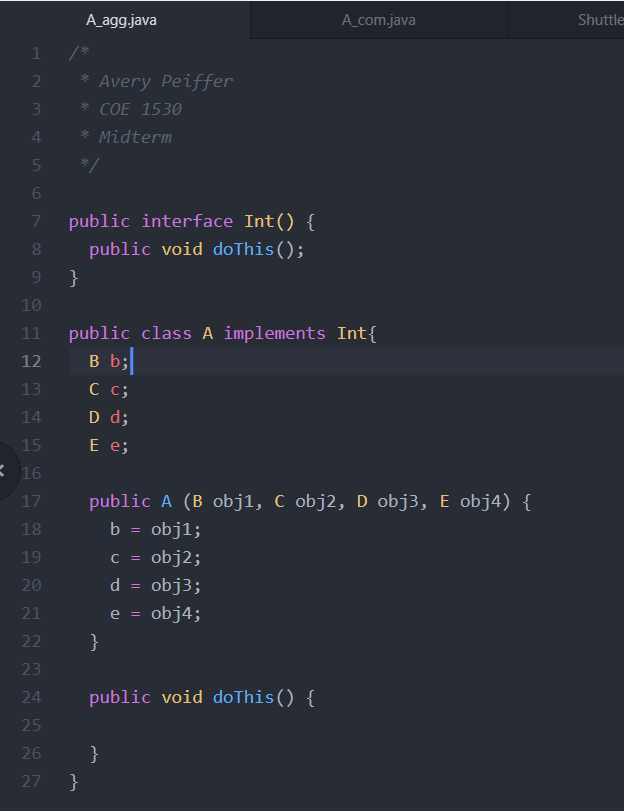


1. Code🡪 UML [20]
   1. Draw a UML class diagram from the following code snippets:[10]





Forgot to put +doThis():void here when drawing

b. Refactor the class A in Question Q4.a above to allow only aggregations (no composition allowed). [5]

c. Refactor the class A in Question Q4.b above to allow only compositions (no aggregations allowed). [5]

