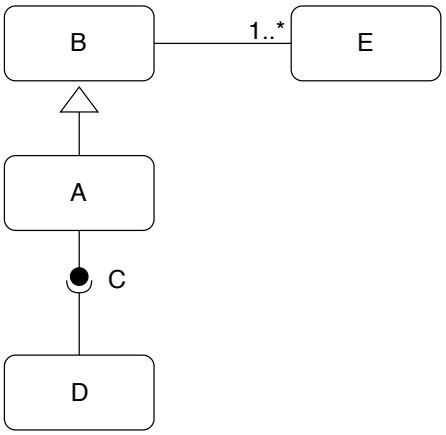
Team: Dieu My Nguyen, Adil Sadik, Yu Li

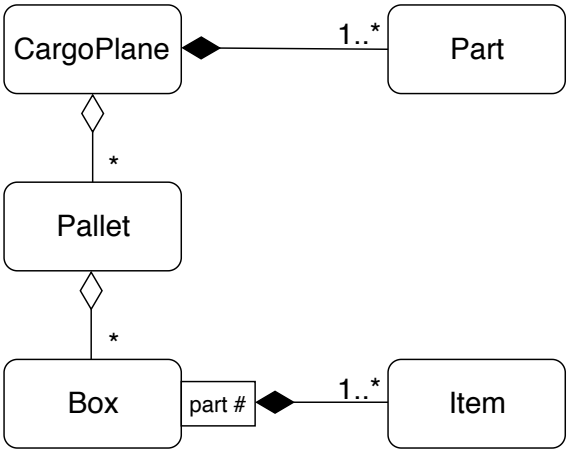
Problem 1

(5 pts) Draw a UML class diagram that models the following statements: "A is a subclass of B. A implements interface C which is used by D to access A. B is associated with one or more E's."



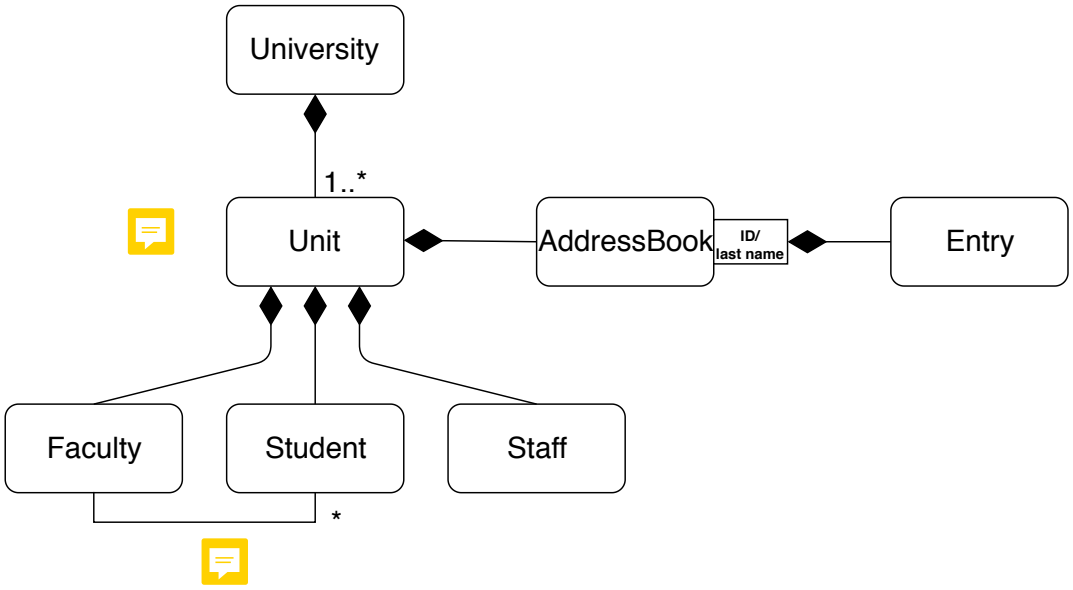
Problem 2

(5 pts) Draw a UML class diagram that models the following statements: "CargoPlane aggregates zero or more Pallets. Each Pallet aggregates zero or more Boxes; CargoPlane is composed of one or more Parts. Inside each Box are one or more Items that are accessed by part number."



Problem 3

(10 pts) Draw a UML class diagram that models the following statements: "A University is composed of one or more Units, such as Colleges and Schools. Each Unit contains faculty, students and staff. A unit maintains an address book filled with entries, one for each type of person contained in that unit. An entry can be located in the address book by supplying their last name or their university id. Faculty members can be the advisor of zero or more students."



Problem 4

(5 pts) Answer the following question about the class diagram you created in question 1: "Can D access an instance of B via C's interface? Explain."

An instance of B cannot be accessed by D via C's interface. D can use C to access instance of A, which contains methods from B (but it is still not an instance of B). However, an instance of B cannot implement A. So, the interface C, which is used to access A instance, cannot be used to access B instance.

Problem 5

(5 pts) Imagine we have a system that has a class called Shape and subclasses Square, Circle, and Triangle each developed by a different software engineer. The system uses these classes to create randomly generated visualizations that balance various constraints concerning the area and perimeter of each individual shape along with the total area and total perimeter

of all shapes on the screen. (An example of such a constraint might be that the total area of all shapes must always be twice as large as the total perimeters.) One engineer, feeling capricious, decides to implement the getArea() method of the Square subclass by having it return the perimeter of the Square instead. What design heuristic or constraint presented in lectures about classes has this engineer violated and what repercussions will it have in the overall system?

The engineer violated the design by contract concept. The subclass Square no longer follows the contract established by the superclass Shape by changing the behavior of the getArea() method. While the system user expects getArea() to output the area, it outputs the perimeter. Consequently, this change will break abstraction and robustness of the overall system.