The case study is an airport parking lot, and we are going to develop the Parking Lot class.

BDD/TDD Version

There will be a stub object of Ticket. Ticket will have three attributes: days, hours and minutes. There will be a Ticket array containing the three values.

The first requirement is to calculate the amount owed by calculate the amount of time parked and applying the rules to calculate the fee.

Parking Lot Fee rules

* The first 30 minutes of the first day are free. (This is in case you pull into the lot and can’t find a space!)
* Time is billed at $2 per hour over that, with a daily max of $15
* Every 7th day is free
* Lost ticket gets charged $75
* We always round time to the nearest hour for charges; Round up for 30 minutes

The second requirement is to print out a report at the end of the week that lists how much money was collected. (for test purposes generate the report when the last car is processed)

Implementation

Input

Read in a file or manually input a file with

* Car ID
* Time in
* Time out
* Lost Ticket Flag (Y/N)

Output

Every time a record is processed.

Output a record with

Car ID

Total Time Parked

Lost Ticket (Y/N)

Fee Collected

Mocking Version

Use mocks to mock out the following;

* Credit card processor service
  + Add a flag to our car record that indicates if the driver is paying with a credit card. If so, then call the credit card processor service (mock it) and also deduct 2.5 percent from the parking fee. The service will return a flag that indicates the transaction is accepted. You can then have it return a transaction denied and then treat the fee as a cash transaction with no deduction
* Fraud System
  + Photo system photos license plate entering and leaving (mock it) .
  + Application logic: If the driver lost his ticket and was in the lot more than 5 days, then the driver is issued with a $150 violation ticket plus a bill for the actual time parked. Set fraud flag and output actual amount plus violation ticket amount