Vending Machine Design

Pre Reflection:

Step to approach this assignment:

Contents

[UML design 1](#_Toc191900808)

[User Story Task 2](#_Toc191900809)

[User story 1: 2](#_Toc191900810)

## UML design

Class: Car, Vending machine

|  |
| --- |
| Class name: Car |
| Attribute/Data member:  String carBranch  String carModel  double carPrice  int carYear  int carFloor  int carSpace |
| Constructors and methods  Car() |

|  |
| --- |
| Class name: Vending Machine |
| Attribute/Data member: |
| Constructors and methods  addCar |

## User Story Task

### User story 1:

**User Story 1:** As a dealership owner, I want to store cars in a vending machine at their designated positions based on reading a file so we can keep track of the cars in the tower.

Scenario:

* Given a vending machine with defined rows and columns entered by the dealer,
* When file is read containing car details,
* *Then the cars are placed into the vending machine at their designated positions.*
  + *Cars are only placed in valid positions within the vending machine.*
  + *A car cannot be placed in an already occupied slot, preventing overwrites.*
  + *Out-of-bounds positions are not allowed*.

addCar : VendingMachine class

readFromFile : driven class

#### TDD

|  |  |  |
| --- | --- | --- |
| Precondition | Postcondition | Comment |
| Vending machine has valid position:  Emty at (1,1). Add car at (1,1) | Car add to valid position  Car store at (1,1) |  |
| Vending machine don’t have valid position:  Has car at (1,2), try to add another car at (1,2) | Can’t not add car |  |
|  |  |  |

#### Method Signature and Algorithms

public void addCar (int floor, int space, Car carToAdd)

if [floor][space] == null

[floor][space] = carToAdd

else Can’t not add car