**Sayuj Shrestha  
CS 2050 (M/W)**Car Vending Machine Design Iteration 02

# Tasks:

User story 1: Load and Store Cars from File

* Create abstract Car class
* Create BasicCar and PremiumCar subclasses
* Make all car fields private/protected with getters
* Implement VendingMachine class with:
* Inventory (LinkedList<Car>)
* Position map (HashMap<String, Car>)
* Update reading car data from file

User story 2: Display Inventory

* Show only filled positions with car details
* Remove empty slot display from original design

User story 3: Sort inventory

* Replace 2D to 1D transformation with sorting of LinkedList directly
* Use comparator instead of insertion sort

User story 4: Retrieve car

* Write new tests

User story 5: Menu Systen

* Implement menu with 10 options in Driver class
* New test scenarios

(New in iteration 02)

User Story 6: Search Cars by Type and Manufacturer

* Implement searchCars(String manufacturer, String type)
* Make it case-insensitive
* New tests

User story 7: Car wash queue system

* Use Queue<Car> to manage car washes
* Add to queue from vending machine
* Process queue in FIFO order

User story 8: Sell a car

* Remove car from inventory
* Make sure that it shows if the slot is empty
* Display “Car Sold: [Details]” on success

# UML Design for class

|  |  |
| --- | --- |
| Classes | Needed |
| Car class (abstract) | - protected int floor  - protected int space  - protected int year  - protected double year  - protected String make  - protected String model  Getters  +getYear()  +getPrice()  +getManufacturer()  +locationKey()  +toString()  +getType() |
| BasicCar | -Extends Car  -Overrides getType() to return “Basic” |
| PremiumCar | -Extends Car  -Overrides getType() to return “Premium” |
| Vending machine Class | -private int floors  -private int spaces  -private LinkedList<Car> inventory  -private Map<String, Car> carPositions  -private Queue<Car> carWashQueue  Methods  +addCar()  +sellCar(int, int)  +displayInventory()  +retrieveCar(int, int)  +printSortedInventory(String)  +addToCarWashQueue(int, int)  +processCarWashQueue() |
| Driver Class | +main method  +readCarFromFile |

# Iteration 01 vs Iteration 02 Method Comparison

Methods that stay the same

|  |  |
| --- | --- |
| **Method** | **Explanation** |
| readCarFromFile() | Still loads car data from a file and constructs Car objects correctly |
| main() | Still manages the overall program flow and menu interaction |

Methods that needed changes

|  |  |
| --- | --- |
| **Method** | **Explanation** |
| addCar() | Now uses Hashmap instead of 2D array |
| retrieveCar() | Now checks the HashMap |
| printSortedInventoryByPrice and Year() | Replaced with a single method printSortedInventory() using Comparator. |
| Car class | Made abstract with common fields/ methods, added getType() to support subclassing. |

New methods adding in iteration 02

|  |  |
| --- | --- |
| **Method** | **Purpose** |
| searchCars() | Enables searching cars by manufacturer and type (Basic/Premium) |
| sellCar() | Removes car from inventory and prints sold confirmation or error |
| addToCarWashQueue() | Adds a retrieve car to queue for washing |
| processCarWashQueue() | Processes cars in FIFO order from the wash queue |

# Unit tests

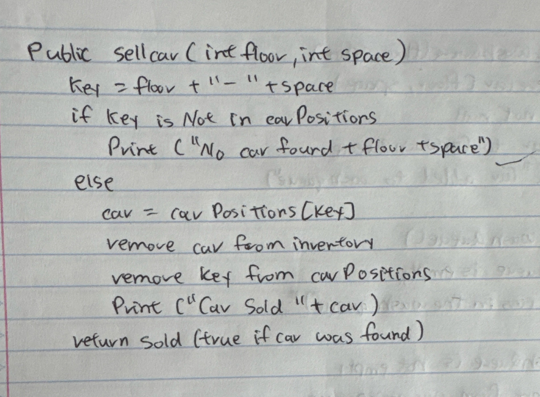
retrieveCar() – HashMap retrieval

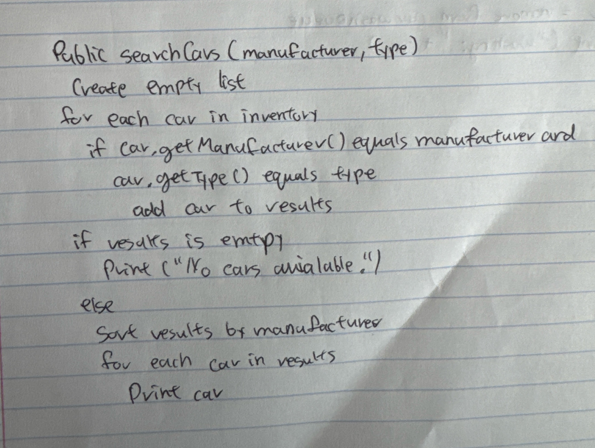
|  |  |  |
| --- | --- | --- |
| **Preconditions** | **Expected Output** | **Notes** |
| Car is added to floor 2, space 3 | Car Retrieved: [car details] | Retrieves from HashMap and prints the correct message |
| No car exists at floor 2, space 3 | Car not found at this location. | Tests behavior when retrieving from an empty or removed location |

searchCars() – Search by Manufacturer and Type

|  |  |  |
| --- | --- | --- |
| **Preconditions** | **Expected Output** | **Notes** |
| Add a Basic Toyota | Shows Toyota car when searching for Toyota + Basic | Must return correct matches based on type and manufacturer |
| Search using lowercase | Still returns correct matches | Search must be case-insensitive |
| Search for a nonexistent manufacturer | No cars available | Should not crash or return incorrect results of not matches are found |

## Pseudocode for the new methods





A close up of a paper

AI-generated content may be incorrect.