Class UsedCarLot

java.lang.Object
    UsedCarLot

```
public class UsedCarLot
extends Object
```

This class is the management of a plethora of car objects all contained within a single "inventory" array.

Author:

Devan Ng

Constructor Summary

Constructors

Constructor	Description
`UsedCarLot()`	Initialization of the car array.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`void`	`addCar(int indexToAdd, Car carToAdd)`	Adds a new car object at the specific index within the inventory array, shifting all cars to the right.
`void`	`addCar(Car newCar)`	Adds another car object the inventory car array.
`ArrayList <Car>`	`getInventory()`	Returns the inventory array holding all the car objects.
`void`	`moveCar(int indexOfCarToMove, int destinationIndex)`	Moves a car object within the inventory array to a different index.
`Car`	`sellCarNoShift (int indexOfCarToSell)`	"Sells" a car object from the inventory array, effectively removing it from the array without shifting the array, instead replacing the sold car object with null.
`Car`	`sellCarShift (int indexOfCarToSell)`	"Sells" a car object from the inventory array, effectively removing it from the

		array and shifting the rest of the car objects in the array to the left.
`boolean`	`swapCar(int car1idx, int car2idx)`	Swaps the car at the first index with the car at the second index of the inventory array.

Methods inherited from class java.lang.Object

`clone , equals , finalize , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait`

Constructor Details

UsedCarLot

`public UsedCarLot()`

Initialization of the car array.

Method Details

getInventory

`public ArrayList <Car> getInventory()`

Returns the inventory array holding all the car objects.

Returns:

The inventory array.

addCar

`public void addCar(Car newCar)`

Adds another car object the inventory car array.

Parameters:

`newCar` - The car object which is being added.

swapCar

```
public boolean swapCar(int car1idx,
                       int car2idx)
```

Swaps the car at the first index with the car at the second index of the inventory array.

Parameters:

`car1idx` - Index of the first car to be swapped.

`car2idx` - Index of the second car to be swapped.

Returns:

either true/false depending on if the swapping of the cars in the array was successful or not.

addCar

```
public void addCar(int indexToAdd,
                   Car carToAdd)
```

Adds a new car object at the specific index within the inventory array, shifting all cars to the right.

PRECONDITION: 0 <= indexToAdd < inventory.size()

Parameters:

`indexToAdd` - The index at which the new car will be added at.

`carToAdd` - The new car object which will be added to the inventory array.

sellCarShift

```
public Car sellCarShift(int indexOfCarToSell)
```

"Sells" a car object from the inventory array, effectively removing it from the array and shifting the rest of the car objects in the array to the left.

PRECONDITION: indexOfCarToSell < inventory.size()

Parameters:

`indexOfCarToSell` - The index of the car to be "sold" and removed from the array.

Returns:

The car object which is being removed.

sellCarNoShift

```
public Car sellCarNoShift(int indexOfCarToSell)
```

"Sells" a car object from the inventory array, effectively removing it from the array without shifting the array, instead replacing the sold car object with null.

PRECONDITION: indexOfCarToSell < inventory.size()

Parameters:

`indexOfCarToSell` - The index of the car to be "sold" and removed from the array.

Returns:

The car object which is being removed.

moveCar

```
public void moveCar(int indexOfCarToMove,
                    int destinationIndex)
```

Moves a car object within the inventory array to a different index.

PRECONDITION: indexOfCarToMove < inventory.size() destinationIndex < inventory.size()

Parameters:

`indexOfCarToMove` - The index of the car object being moved.

`destinationIndex` - The index of where to put the car object being moved.