Class UsedCarLot

java.lang.Object
    UsedCarLot

```
public class UsedCarLot
extends Object
```

Constructor Summary

Constructors

Constructor	Description
`UsedCarLot()`	Instantiates a usedCarLot object.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
void	`addCar(int indexToAdd, Car carToAdd)`	Adds a Car to inventory at the given index
void	`addCar(Car car)`	Adds new Car and relative information to inventory
`ArrayList <Car>`	`getInventory()`	Gets the inventory of all values added ot the list.
void	`moveCar(int indexOfCarToMove, int destinationIndex)`	Moves car and and shifts array based off of new position
Car	`sellCarNoShift (int indexOfCarToSell)`	Sells car and replaces value with null
Car	`sellCarShift (int indexOfCarToSell)`	Sells car and removes from the inventory list
boolean	`swap(int first, int second)`	Switches placement of two values given two indexes.

Methods inherited from class java.lang.Object

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

Constructor Details

UsedCarLot

```
public UsedCarLot()
```

Instantiates a usedCarLot object.

Method Details

getInventory

```
public ArrayList <Car> getInventory()
```

Gets the inventory of all values added ot the list.

Returns:

inventory list

addCar

```
public void addCar(Car car)
```

Adds new Car and relative information to inventory

swap

```
public boolean swap(int first,
                    int second)
```

Switches placement of two values given two indexes.

If both values are greater than 0. Finds index of first value and sets to temp; Sets value at index first to the value at index second Sets value at inde second to temp

Parameters:

`first` - is one of the values index you want to switch

`second` - is one of the values index you want to switch

Returns:

true if everything was executed, false if not

addCar

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

Adds a Car to inventory at the given index

PRECONDITION: indexToAdd is within range of list

Parameters:

`indexToAdd` - where to add the given Car value

`carToAdd` - Car value to add

sellCarShift

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

Sells car and removes from the inventory list

PRECONDITION: indexOfCarToSell is less than size of inventory

Parameters:

`indexOfCarToSell` - index of what value to remove

Returns:

value removed

sellCarNoShift

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

Sells car and replaces value with null

PRECONDITION: indexOfCarToSell is less than size of inventory

Parameters:

`indexOfCarToSell` - index of what value to remove

Returns:

sold car with updated value of null

moveCar

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

Moves car and and shifts array based off of new position

PRECONDITION: indexOfCarToMove and destinationIndex is less than size of inventory

Parameters:

`indexOfCarToMove` - index of what Car to move

`destinationIndex` - index of where Car should be moved