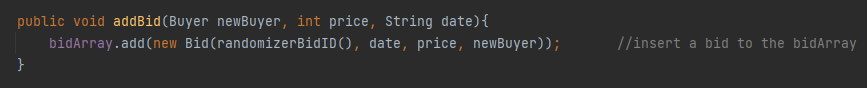


Right now, for any given car object, their bid data and buyer data are stuck together.



To add an element in the bidArray for a car, they would need a new buyer object, so when functions like displayCars calls this bidArray, it will receive the buyer’s info on top of the bid’s info.

Text

Description automatically generated

Not only will you only be able to display both infos, but they also have no filter, buyers with the exact same information will be displayed as two different things, even if their bids and bid date is the same, it will still show up as two things when displayed.

One way to extract **only** the buyer info is to extract it from the bidArray, to call bidArray but only take the buyer information and leave out the bidding info.Text

Description automatically generated

A similar loop can be made like in the displayCars method, but instead of taking everything (car info, buyer infos, and bidder infos), it can instead only take the buyer info.

To filter out any duplicates we could have two temporary buyer infos one that has the info from the last index and the other one has the info loop is currently processing.

so the code for the method may go like this…

*Public void displayBuyers(){*

*Access Buyer info from bidArray*

*For (index < amount of buyers in one car){*

*If(index =! 0){ //make sure no errors is made from index trying to go to -1*

*tempLastBuyer = buyer from index-1*

*}*

*tempBuyer = buyer from index*

*if(tempBuyer == tempLastBuyer){*

*skip to next buyer*

*}else{*

*Sout(buyer description)*

*}*

*}*