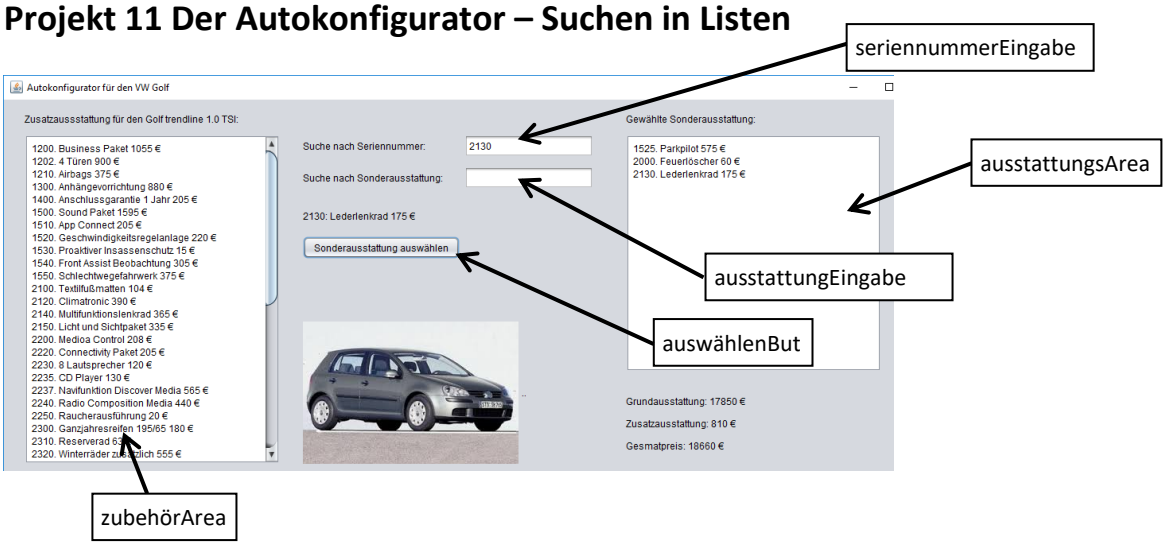
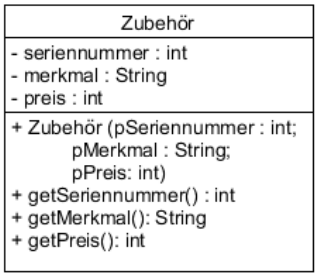
Projekt 11 Der Autokonfigurator – Suchen in Listen



Beim Kauf eines neuen Autos ist eine App sehr hilfreich, mit der man das Zubehör für das neue Gefährt zusammenstellen kann. Eine App soll für das Beispiel eines VW Golf trendline erstellt werden. Kopiere dazu das Projekt „Autokonfigurator“ aus dem Schülerverzeichnis. In der **zubehörArea** werden verschiedene Ausstattungen aufgelistet, die der Benutzer auswählen kann. Dazu gibt man in der **JTextArea seriennummerEingabe** die Seriennummer ein. Die App soll in der Liste aller Seriennummern die gewählte Nummer suchen und anschließend das ausgewählte Zubehör aus der **zubehörListe** entfernen sowie diese in die neue Liste der gewählten Ausstattungen (**ausstattungsListe**) aufnehmen.

zubehörListe und ausstattungsListe sind Listen vom Typ **Zubehör**, für die eine eigene Klasse bereits angelegt ist. Die Klasse Zubehör enthält die drei Attribute **seriennummer**, **merkmal** und **preis**. Dabei ist merkmal der passende Text des Zubehörs, z. B. „Business Paket“. Die Klasse Zubehör ist fertig implementiert entsprechende dem rechts abgebildeten Implementationsdiagramm.	

Beim Start der App wird die **zubehörListe** bereits in der Methode **Aufbauen()** fertig aufgebaut. Die Methode **listeAusgeben (List<Zubehör> liste, JTextArea area1)** kann die **zubehörListe** oder die **ausstattungsListe** in einer der beiden **JTextAreas** bereits fertig ausgeben. Wenn du das Projekt startest, siehst du, wie die linke **JTextArea** mit allen verfügbaren Ausstattungsmöglichkeiten aufgefüllt wird.

Aufgabe 1: Wenn der Benutzer in der **JTextArea seriennummerEingabe** eine Seriennummer eingibt, soll die App in der **zubehörListe** suchen, ob es ein Zubehör mit der eingegebenen Seriennummer gibt. Wenn ja, soll zunächst im **ergebnisLab** das Zubehör vollständig, d. h. mit Seriennummer, Merkmal und Preis ausgegeben werden. Später soll dieses Zubehör –wie oben schon gesagt - in der Ausstattungsliste eingefügt werden.

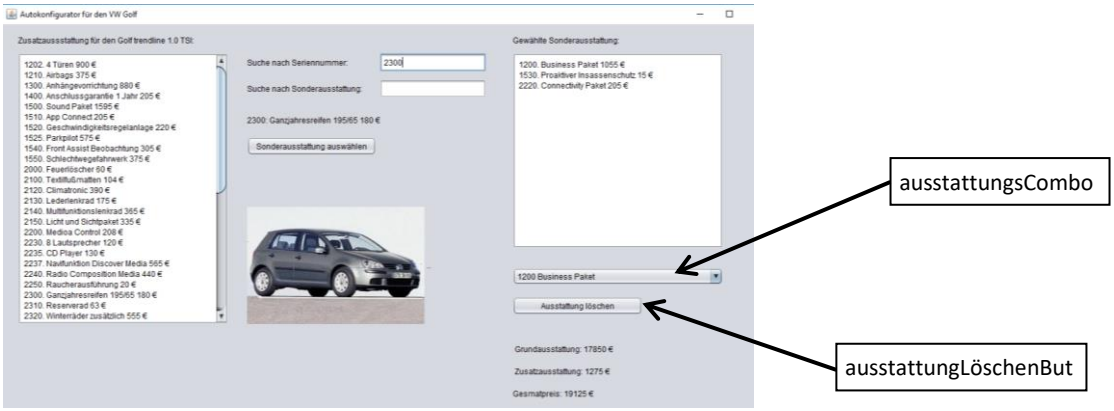
Der rechts angegebene Pseudocode zeigt, wie man in einer Liste nach einer vorgegebenen Seriennummer sucht. Implementiere die entsprechende Methode **seriennummerEingabeKeyReleased**, die bei Eingabe im **JTextField seriennummerEingabe** nach jedem Tastendruck ausgeführt wird. Um festzustellen, wann die Suche erfolgreich beendet wurde verwendet man am besten eine boolsche Variable **fertig**, die zu Beginn der Suche auf **false** gesetzt wird. Am Ende der Suche steht das aktuelle Element der **zubehörListe** (current) an der Stelle des gesuchten Zubehörs. Das kann später benutzt werden, um das Zubehör in die **ausstattungsListe** zu übernehmen, die in der rechten TextArea dargestellt wird.	gehe an den Anfang der Liste wiederhole solange die Liste ein aktuelles Element hat und das gesuchte Element noch nicht gefunden wurde wenn die Seriennummer des aktuellen Elementes gleich dem Inhalt des Textfeldes ist markiere die Suche als gefunden wenn nicht gehe zum nächsten Element ende wenn ende wiederhole wenn die Suche erfolgreich war gebe das gefundene Zubehör aus wenn nicht gebe aus, dass die Suche nicht erfolgreich war ende wenn

Aufgabe 2: Ganz analog soll die Suche nach einem bestimmten Merkmal implementiert werden. Wenn der Benutzer z. B. im **JTextField merkmalEingabe** die Eingabe „Airbags“ macht, soll das entsprechende Zubehör gesucht werden. Der Methodeninhalt kann im Wesentlichen kopiert werden, beim Vergleich von Strings musst du daran denken, dass die equal-Methode benutzt werden muss.

Aufgabe 3: Wenn durch die bisherigen Methoden ein Zubehör gefunden wurde und der Benutzer anschließend auf den Button **auswählenBut** klickt, soll das gesuchte Merkmal aus der **zubehörListe** (Liste aller verfügbaren Zubehöre, links dargestellt) entfernt und in die **ausstattungsListe** (ausgewählte Zubehöre, rechts dargestellt) eingefügt werden. Anschließend sollen beide Listen mit der vorhandenen Methode am Bildschirm neu dargestellt werden. Implementiere die Methode **auswählenButMouseClicked**, die auf den entsprechenden Button reagiert.

Zusatzaufgabe 1: Im Bereich unten rechts soll der bisherige Gesamtpreis des Autos nach jeder Auswahl eines Zubehörs aktualisiert werden. Implementiere die notwendigen Schritte. Eine Methode **preisberechnen()** ist bereits zum Teil implementiert.

Zusatzaufgabe 2: Es soll möglich werden, ein ausgewähltes Zubehör wieder aus der **ausstattungsListe** zu entfernen. Schreibe dazu zunächst eine Methode **ausstauungsComboSetzen()**, die alle aktuell ausgewählten Ausstattungen der **ausstattungsliste** in die **ausstattungsCombo** überträgt. Zum Setzen eines Items in der **JCombobox ausstattungsCombo** benötigst du den Befehl **ausstattungsCombo.addItem („Text“)**. Als Text reicht hier die Angabe von Seriennummer und Merkmal. Zu Beginn muss die Combo mit dem Befehl **ausstattungsCombo.removeAllItems();** gelöscht werden. Die Methode muss am Ende der Methode aufgerufen werden, die auf den **auswählenBut** reagiert, damit die Combobox immer alle aktuell ausgewählten Ausstattungen enthält. Anschließend muss die Methode **ausstattungLöschenButMouseClicked** implementiert werden: Nach Klick auf den Button muss der Eintrag in der **ausstattungsListe** entfernt werden. Dazu kann man mit dem Befehl **int pos1 = ausstattungsCombo.getSelectedIndex();** zunächst herausfinden, das wievielte Element aus der **ausstattungsListe** entfernt werden muss. Anschließend durchläuft man die **ausstattungsListe** bis zu diesem Element und löscht es. Zum Schluss sollte das gelöschte Element in der **zubehörliste** am Ende wieder eingefügt werden.



Ergänzungen zur Zusatzaufgabe 2 in der Gui