

Autos

Schreibe ein Programm, das eine Liste von Autos aus einer Datei einliest. Jede Zeile beschreibt ein Auto und hat das Format **<brand> <speed> <color>**. Implementiere drei Funktionen:

void printCars(std::vector<Car> & cars) Bekommt eine Referenz auf einen Vektor von **Car** Objekten und gibt alle Autos in der Form **<brand>: <speed> (<color>)\n** aus.

Beispiel

```
Audi: 280 (black)
Audi 280 silver
VW 220 silver
```

void sortCars(std::vector<Car> & cars) Bekommt eine Referenz auf einen Vektor von **Car** Objekten und sortiert diese absteigend nach Geschwindigkeit.

Beispiel

```
Audi: 330 (red)
BMW: 310 (white)
BMW: 310 (black)
```

void removeDuplicateCars(std::vector<Car> & cars) Bekommt eine Referenz auf einen Vektor von **Car** Objekten und entfernt alle doppelten Autos. Autos sind gleich, wenn sowohl die Marke (**brand**) als auch die Geschwindigkeit (**speed**) übereinstimmen. (Es ist in Ordnung, wenn die Autos dabei auch sortiert werden.)

Beispiel

```
Audi: 330 (red)
BMW: 310 (white)
Audi: 280 (black)
```

Ist die Anzahl der Parameter falsch, soll "Usage: ./cars <car file>" ausgegeben werden. Sollte eine Zeile in der Datei ungültig sein, soll sie ignoriert werden.

Beispiel für eine Auto-Liste **cars.txt**:

```
Audi 280 black
Audi 280 silver
BMW 310 white
VW 220 silver
Audi 330 red
VW 180 blue
BMW 310 black
VW 280 silver
```

Aufruf

./cars <car list>

Bewertung

- Dokumentation und Programmierstil: 1 Punkt
- Einhalten der OOP Konzepte: 1 Punkt
- Struktur & Korrektheit: 1 Punkt