OPGAVE  
Cars sorted by plate

I denne opgave træner vi enkel sortering med Comparable.

**Car-klassen**

1. Opret klassen Car med attributten plate (nummerplade), som har datatypen String.
2. Opret en konstruktør, der modtager plate som parameter.
3. Lad Car implementere Comparable<Car>.
4. Tilføj metoden compareTo (IntelliJ foreslår det selv, så du kan bare acceptere forslaget). Lad være med at ændre i koden – bare acceptér return 0; foreløbig.
5. Opret en getter til plate.

**SortCars**

1. Opret klassen SortCars med main -> run. Arbejd videre i run-metoden nedenfor.
2. Opret en ArrayList<Car> kaldet cars.
3. Opret et antal biler og tilføj dem i listen.
4. Udskriv listen på skærmen (brug getPlate).
5. Kald Collections.sort(cars).
6. Udskriv listen på skærmen igen.

Programmet virker, men de to lister er ens. Til sidst skal vi have sorteringen til at fungere:

**compareTo()**

1. Gå tilbage til compareTo() og lav den om, så metoden sorterer efter plate:

return this.plate.compareTo(o.plate);

1. Kør programmet igen. Bliver listen ordnet i stigende eller faldende orden? Prøv at ændre det, så programmet gør det modsatte.

Eksempel på bil data i gerne m å bruge:

ArrayList<Car> cars = new ArrayList<>();  
  
cars.add(new Car("Subaru", 2015, "Blue", 67000));  
cars.add(new Car("Kia", 2018, "Red", 36000));  
cars.add(new Car("Mazda", 2011, "Silver", 89000));  
cars.add(new Car("Chevrolet", 2014, "Black", 52000));  
cars.add(new Car("Fiat", 2009, "White", 110000));  
cars.add(new Car("Peugeot", 2016, "Green", 45000));  
cars.add(new Car("Opel", 2013, "Grey", 60000));  
cars.add(new Car("Citroen", 2012, "Yellow", 72000));  
cars.add(new Car("Dacia", 2017, "Orange", 38000));  
cars.add(new Car("Suzuki", 2010, "Brown", 80000));  
cars.add(new Car("Lada", 2005, "Purple", 120000));  
cars.add(new Car("Smart", 2019, "Pink", 28000));  
cars.add(new Car("Chrysler", 2008, "Gold", 95000));  
cars.add(new Car("Jaguar", 2014, "Cyan", 54000));  
cars.add(new Car("Lamborghini", 2020, "Lime", 18000));  
cars.add(new Car("Bugatti", 2016, "Teal", 47000));  
cars.add(new Car("Ferrari", 2015, "Magenta", 59000));  
cars.add(new Car("Porsche", 2017, "Turquoise", 42000));  
cars.add(new Car("Bentley", 2018, "Maroon", 34000));  
cars.add(new Car("Rolls-Royce", 2021, "Beige", 12000));  
cars.add(new Car("McLaren", 2019, "Violet", 25000));  
cars.add(new Car("Aston Martin", 2019, "Indigo", 26000));  
cars.add(new Car("Lotus", 2014, "Crimson", 56000));  
cars.add(new Car("Mini", 2016, "Olive", 48000));  
cars.add(new Car("Tesla", 2022, "Plum", 5000));