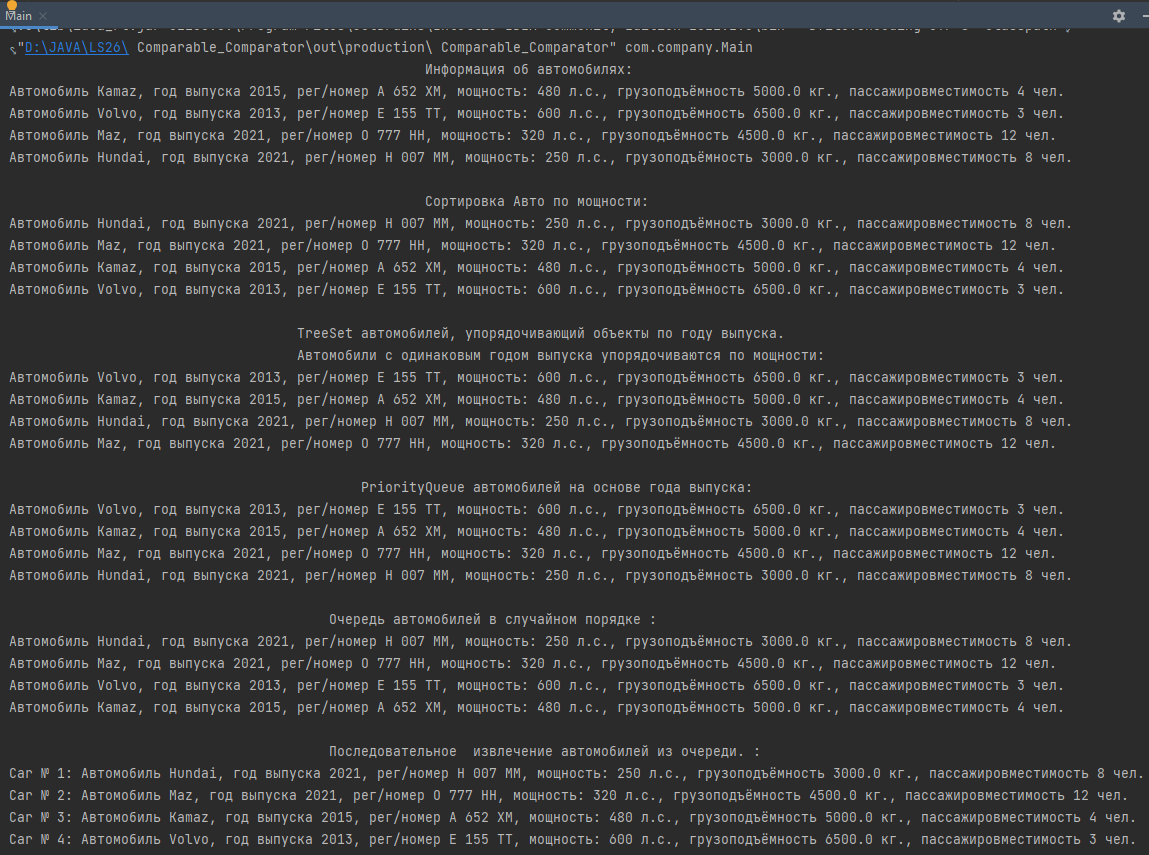
***class Car implements Comparable<Car>***

package com.company;  
  
import java.util.Comparator;  
  
*/\*\*  
 \* Задание.  
 \* 1. Реализовать класс «Автомобиль». Автомобиль имеет гос номер, мощность л.с., год выпуска, грузоподъёмность,  
 \* кол-во пассажирских мест.  
 \* Сделать конструктор (конструкторы) с параметрами, toString и все прочие необходимые методы для данного класса.  
 \* 2. Для класса из п.1. выполнить задания:  
 \* • Создать список автомобилей, вывести, отсортировать по мощности, вывести  
 \* • Создать TreeSet автомобилей, упорядочивающий объекты по году выпуска. Автомобили с одинаковым годом выпуска  
 \* упорядочиваются по мощности.  
 \* • Создать PriorityQueue автомобилей на основе года выпуска. Добавить в очередь автомобили в случайном порядке,  
 \* затем последовательно извлекать их из очереди и выводить.  
 \* В программе должны быть имплементации интерфейсов Comparable и Comparator – обязательно. Stream API, lambda,  
 \* анонимные классы использовать запрещено.  
 \*/*// класс "Автомобиль"  
public class Car implements Comparable<Car> {  
  
 // поля класса "Автомобиль"  
 private String name; //название авто  
 private String regNum; //регистрационный номер  
 private int power; //мощность двигателя л.с.  
 private int yearProduct; //год выпуска автомобиля  
 private double loadCapacity; //грузоподъёмность  
 private int passengerCapacity; //кол-во пассажирских мест  
  
 // конструкторы  
 // 1. конструктор без параметров - по умолчанию  
 public Car() {  
 // инициализация полей по умолчанию  
 name = " \"Car\"";  
 regNum = "A 000 AA 54 RUS";  
 power = 80;  
 yearProduct = 1987;  
  
 }  
  
 // 2. конструктор с 6 параметрами  
 public Car(String *name*, String *regNum*, int *power*, int *yearProduct*,  
 double *loadCapacity*, int *passengerCapacity*) {  
 this.name = *name*;  
 this.regNum = *regNum*;  
 this.power = *power*;  
 this.yearProduct = *yearProduct*;  
 this.loadCapacity = *loadCapacity*;  
 this.passengerCapacity = *passengerCapacity*;  
 }  
  
 // методы  
 // 1. метод представления класса в виде строки - перегрузка метода toString  
 @Override  
 public String toString() {  
 return "Автомобиль " + name + ", год выпуска " + yearProduct + ", рег/номер " +  
 regNum + ", мощность: " + power + " л.с., " + "грузоподъёмность " + loadCapacity +  
 " кг., пассажировместимость " + passengerCapacity + " чел.\n";  
 }  
  
 // 2. метод compare - сравнивает автомобили по мощности  
 @Override  
 public int compareTo(Car *auto*) {  
 //сравниваем авто по мощности power  
 return Integer.*compare*(power, *auto*.power);  
 }  
  
 //класс компаратора  
 public static class ComparatorCar implements Comparator<Car> {  
  
 //метод сравнения по году выпуска, если год выпуска одинаковый, сравнивает по мощности  
 @Override  
 public int compare(Car *auto\_1*, Car *auto\_2*) {  
 if (*auto\_1*.yearProduct > *auto\_2*.yearProduct) {  
  
 return 1;  
 } else if (*auto\_1*.yearProduct < *auto\_2*.yearProduct) {  
 return -1;  
 } else return Double.*compare*(*auto\_1*.power, *auto\_2*.power);  
 }  
  
 }//class ComparatorCar close  
  
}//class Car close

***class Main***

package com.company;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.PriorityQueue;  
import java.util.TreeSet;  
  
public class Main {  
  
 public static void main(String[] *args*) {  
 //Создали список автомобилей  
 ArrayList<Car> *auto\_1* = new ArrayList<>();  
  
 //Добавили в список автомобили  
 *auto\_1*.add(new Car("Kamaz", "А 652 ХМ", 480, 2015, 5000, 4));  
 *auto\_1*.add(new Car("Volvo", "Е 155 ТТ", 600, 2013, 6500, 3));  
 *auto\_1*.add(new Car("Мaz", "О 777 НН", 320, 2021, 4500, 12));  
 *auto\_1*.add(new Car("Hundai", "H 007 ММ", 250, 2021, 3000, 8));  
  
 //вывод данных автомобилей на экран  
 System.*out*.println("\t\t\t\t\t\t\t\t\t\t\t\t\tИнформация об автомобилях:");  
 for (Car *item* : *auto\_1*) {  
 System.*out*.print(*item*);  
 }  
 System.*out*.println();  
  
 Collections.*sort*(*auto\_1*); //сортировка автомобилей по мощности  
 System.*out*.println("\t\t\t\t\t\t\t\t\t\t\t\t\tСортировка Авто по мощности:");  
 //вывод отсортитрованных авто на экран  
 for (Car *item* : *auto\_1*) {  
 System.*out*.print(*item*);  
 }  
 System.*out*.println();  
  
 //TreeSet автомобилей, упорядочивающий объекты по году выпуска.  
 // Автомобили с одинаковым годом выпуска упорядочиваются по мощности.  
 //Создали TreeSet автомобилей  
 TreeSet<Car> *auto\_2* = new TreeSet<>(new Car.ComparatorCar());  
  
 //Добавили в TreeSet автомобили  
 *auto\_2*.add(new Car("Kamaz", "А 652 ХМ", 480, 2015, 5000, 4));  
 *auto\_2*.add(new Car("Volvo", "Е 155 ТТ", 600, 2013, 6500, 3));  
 *auto\_2*.add(new Car("Мaz", "О 777 НН", 320, 2021, 4500, 12));  
 *auto\_2*.add(new Car("Hundai", "H 007 ММ", 250, 2021, 3000, 8));  
  
 System.*out*.println("\t\t\t\t\t\t\t\t\tTreeSet автомобилей, упорядочивающий объекты по году выпуска.");  
 System.*out*.println("\t\t\t\t\t\t\t\t\tАвтомобили с одинаковым годом выпуска упорядочиваются по мощности:");  
 //вывод данных автомобилей на экран  
 for (Car *item* : *auto\_2*) {  
 System.*out*.print(*item*);  
 }  
 System.*out*.println();  
  
 //PriorityQueue очередь автомобилей на основе года выпуска.  
 //Создали PriorityQueue очередь автомобилей  
 PriorityQueue<Car> *auto\_3* = new PriorityQueue<>(4, new Car.ComparatorCar());  
  
 //Добавили в PriorityQueue автомобили  
 *auto\_3*.add(new Car("Kamaz", "А 652 ХМ", 480, 2015, 5000, 4));  
 *auto\_3*.add(new Car("Volvo", "Е 155 ТТ", 600, 2013, 6500, 3));  
 *auto\_3*.add(new Car("Мaz", "О 777 НН", 320, 2021, 4500, 12));  
 *auto\_3*.add(new Car("Hundai", "H 007 ММ", 250, 2021, 3000, 8));  
  
 System.*out*.println("\t\t\t\t\t\t\t\t\t\t\tPriorityQueue автомобилей на основе года выпуска:");  
 //вывод данных автомобилей на экран  
 for (Car *item* : *auto\_3*) {  
 System.*out*.print(*item*);  
 }  
 System.*out*.println();  
  
 System.*out*.println("\t\t\t\t\t\t\t\t\t\tОчередь автомобилей в случайном порядке :");  
 //Создали PriorityQueue очередь автомобилей  
 PriorityQueue<Car> *auto\_4* = new PriorityQueue<>();  
  
 //Добавили в PriorityQueue автомобили в случайном порядке  
 *auto\_4*.add(new Car("Hundai", "H 007 ММ", 250, 2021, 3000, 8));  
 *auto\_4*.add(new Car("Kamaz", "А 652 ХМ", 480, 2015, 5000, 4));  
 *auto\_4*.add(new Car("Volvo", "Е 155 ТТ", 600, 2013, 6500, 3));  
 *auto\_4*.add(new Car("Мaz", "О 777 НН", 320, 2021, 4500, 12));  
  
 //вывод данных автомобилей на экран  
 for (Car *item* : *auto\_4*) {  
 System.*out*.print(*item*);  
 }  
 System.*out*.println();  
  
 System.*out*.println("\t\t\t\t\t\t\t\t\t\tПоследовательное извлечение автомобилей из очереди. :");  
 // Последовательное извлечение автомобилей из очереди  
 int *count* = *auto\_4*.size();  
 for (int *i* = 0; *i* < *count*; *i*++) {  
 //выводит элемент  
 Car *item* = *auto\_4*.element();  
 //извлекает элемент  
 *auto\_4*.remove();  
 //вывод элемента  
 System.*out*.print("Car № " + (*i* + 1) + ": " + *item*);  
 }  
  
 }//main  
  
}//class close

***Тесты:***

******