Класс CarParking

```java
package package3;

public class CarParking {
    public int maxCarsNumber = 0;
    private CarPlace[] carPlaces;
    private int counter = 0;
    public CarParking(int maxCarsNumber)
    {

        this.maxCarsNumber = maxCarsNumber;

        carPlaces = new CarPlace[this.maxCarsNumber];

        for(int i = 0; i < maxCarsNumber; i++){
            carPlaces[i] = new CarPlace(i+1);
        }
    }

    public void addCar(Car car)
    {
        if(counter < maxCarsNumber){
            carPlaces[counter].vehicle = car;
            counter++;
        }
        else
            System.out.println("Все места заняты!");

    }

    public void sortByEngine()
    {
        //Сортировка "пузырьком"
        CarPlace tempCarPlace;
            for(int i = 0; i < maxCarsNumber; i++)
            {
                for(int j=maxCarsNumber-1;j>i;j--)
                {
                    if(carPlaces[j-1].vehicle.getEngine() > carPlaces[j].vehicle.getEngine())
                    {
```

```
39                  {
40                      tempCarPlace = carPlaces[j-1];
41                      carPlaces[j-1] = carPlaces[j];
42                      carPlaces[j] = tempCarPlace;
43                  }
44              }
45          }
46      }
47
48      public void sortById()
49      {
50          //Сортировка выбором
51          int minValueIndex ;
52          CarPlace tempPlace;
53          for(int i = 0; i< maxCarsNumber; i++)
54          {
55              minValueIndex = i; // Выбираем за минимальное значение первый не сортированный элемент
56              for(int j = i+1;j < maxCarsNumber; j++ )
57              {
58                  if(carPlaces[j].id<carPlaces[minValueIndex].id) // Если элемент меньше принятого за наименьшее
59                  {
60                      minValueIndex = j;                          // то меняем индекс наименьшего значения
61                  }
62              }
63              tempPlace = carPlaces[i];                // Устанавливаем найденное значение под индексом i
64              carPlaces[i] = carPlaces[minValueIndex];
65              carPlaces[minValueIndex] = tempPlace;
66          }
67      }
68
69      @Override
70      public String toString()
71      {
72          String result = "";
73          for(int i = 0; i < maxCarsNumber; i++)
74          {
75              if(null != carPlaces[i].vehicle)
76                  result+=carPlaces[i].vehicle.toString()+ "; Номер места: " + carPlaces[i].id +"\n";
77              else
```

```
77              else
                    result+="Пусто!";
79          }
80          return result;
81      }
82  }
83
```

Класс BusParking

```java
package package3;

import java.util.ArrayList;


public class BusParking {
    public int maxBusesNumber = 0;
    private ArrayList<BusPlace> busPlaces;
    private int counter = 0;

    public BusParking(int maxBusesNumber)
    {

        this.maxBusesNumber = maxBusesNumber;

        busPlaces = new ArrayList<BusPlace>();


    }

    public void addBus(Bus bus)
    {
        if(busPlaces.size() < maxBusesNumber)
        {
            BusPlace tempBusPlace = new BusPlace(counter+1);
            tempBusPlace.vehicle=bus;
            busPlaces.add(tempBusPlace);
            counter++;

        }
    }

    public void sortByEngine()
    {
        //Сортировка пузырьком
        BusPlace tempBusPlace;
        for(int i = 0; i < maxBusesNumber; i++) //выполняем n проходов по массиву, где n-размер массива
        {
            for(int j = maxBusesNumber-1 ; j > i ; j--) //проходим по массиву и сравниваем соседние элементы
```

```
39              for(int j = maxBusesNumber-1 ; j > i ; j--) //проходим по массиву и сравниваем соседние элемент
40              {
41                  if(busPlaces.get(j-1).vehicle.getEngine() > busPlaces.get(j).vehicle.getEngine())
42                  {
43                      //Если элемент j больше чем j+1, то меняем их местами
44                      tempBusPlace = busPlaces.get(j-1);
45                      busPlaces.set(j-1,  busPlaces.get(j));
46                      busPlaces.set(j,tempBusPlace);
47                  }
48              }
49          }
50      }
51
52      public void sortById()
53      {
54          //Сортировка выбором
55          int minValueIndex ;
56          BusPlace tempPlace;
57          for(int i = 0; i< maxBusesNumber; i++)
58          {
59              minValueIndex = i; // Выбираем за минимальное значение первый не сортированный элемент
60              for(int j = i+1;j < maxBusesNumber; j++ )
61              {
62                  // Если элемент меньше принятого за наименьшее значение,
63                  if(busPlaces.get(j).id<busPlaces.get(minValueIndex).id)
64                  {
65                      minValueIndex = j;                          // то меняем индекс наименьшего значения
66                  }
67              }
68              tempPlace = busPlaces.get(i);               // Устанавливаем найденное значение под индексом i
69              busPlaces.set(i, busPlaces.get(minValueIndex));
70              busPlaces.set(minValueIndex, tempPlace);
71          }
72      }
73
75      @Override
76      public String toString()
77      {
          String result = "";
```

```
77          String result = "";
78          for(int i = 0; i < busPlaces.size(); i++)
79          {
80              if(busPlaces.get(i).vehicle != null )
                    result+=busPlaces.get(i).vehicle.toString()+ "; Номер места: " + busPlaces.get(i).id +"\n";
82              else
                    result+="Пусто!";
84          }
85          return result;
86      }
87  }
```

Класс Vehicle

```java
package package3;
//реализация интерфейса Comparable, используется для сравния двух объектов.
//Будет использоваться в пакете с коллекциями.
public class Vehicle implements Comparable{
    private String name;
    private String color;
    private int engine;
    private String gosNumber;

    public Vehicle(String name,String color,String gosNumber,int engine)
    {
        this.name=name;
        this.color=color;
        this.engine=engine;
        this.gosNumber=gosNumber;
    }
    public String toString()
    {
        return "Трасннортное средство "+getName()+" цвет:"+getColor()+", государственный номер: "
                +getGosNumber()+", двигатель "+getEngine()+" лс";
    }

   //Инкапсуляция
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getColor() {
        return color;
    }
    public void setColor(String color) {
        this.color = color;
    }
    public int getEngine() {
        return engine;
    }
    public void setEngine(int engine) {
```

```
39      public void setEngine(int engine) {
40          this.engine = engine;
41      }
42      public String getGosNumber() {
43          return gosNumber;
44      }
45      public void setGosNumber(String gosNumber) {
46          this.gosNumber = gosNumber;
47      }
48
49      @Override
50      public int compareTo(Object o) {
51          // Сравнение двух объектов. Если они равны, то метод возвращает 0
52          // Если данный объект "больше" чем сравниваемый, то метод вернет положительный результат
53          // Наоборот-отрицательный
54          if(this.gosNumber.equals(((Vehicle)o).getGosNumber()))
55          {
56              return 0;
57          }
58          return this.gosNumber.compareTo(((Vehicle)o).getGosNumber());
59      }
60      @Override
61      public int hashCode()
62      {
63          return this.hashCode();
64      }
65      @Override
66      public boolean equals(Object o)
67      {
68          Vehicle v=(Vehicle) o;    //Приведение типов
69          if(this.name.equals(v.getName()) && this.gosNumber.equals(v.getGosNumber()))
70          {
71               return true;
72          }
73          return false;
74
75      }
76  }
```

Класс Main

```java
package package3;

public class main {
        public static void main(String[] args) {
        // CarParking основанный на массиве
        CarParking cp = new CarParking(3);

        Car car=new Car("Chevrolet","Белый","95",245,"sedan");
        Car car2=new Car("Chevrolet2","Белый","95",210,"sedan");
        Car car3=new Car("Chevrolet3","Белый","95",231,"sedan");

        cp.addCar(car);
        cp.addCar(car2);
        cp.addCar(car3);

        System.out.println(cp);

        cp.sortByEngine();

        System.out.println("Отсортированный по мощности двигателя массив :");
        System.out.println(cp);

        cp.sortById();

        System.out.println("Отсортированный по номеру парковочного места массив :");
        System.out.println(cp);

        // BusParking основан на ArrayList
        System.out.println("//-------------------------------------------------//");

        Bus ikarus=new Bus("Ikarus","Зеленый","v543gg",192,165F,37);
        Bus gazel=new Bus("Gazel","Красный","v783fg",152,4.8F,6);
        Bus max = new Bus("Maz","Белый","v346rt",157,143F,30);


        BusParking bp = new BusParking(3);
        bp.addBus(max);
        bp.addBus(gazel);
        bp.addBus(ikarus);
```

```
39            bp.addBus(ikarus);
40
41
42            System.out.println(bp);
43
44            bp.sortByEngine();
45
46            System.out.println("Отсортированный по мощности двигателя массив :");
47            System.out.println(bp);
48
49            bp.sortById();
50
51            System.out.println("Отсортированный по номеру парковочного места массив :");
52            System.out.println(bp);
53
54        }
55    }
```