

## 3.6

### Разбираем сложный код

00:00—00:30

Привет! В этом видеоматериале мы объединим полученные знания по синтаксису языка Java и разберём код консольного приложения, в котором содержатся сразу несколько различных операторов.

Если вы начинаете работать в новой компании, то вам, как правило, приходится работать с уже имеющимся кодом и поэтому разработчики должны уметь не только писать новый код, но и читать тот, который уже написан.

00:30—04:38

```
1  import java.util.Scanner;
2
3  ▶ public class RoadController {
4      private static double passengerCarMaxWeight = 3500.0; // kg
5      private static int passengerCarMaxHeight = 2000; // mm
6      private static int controllerMaxHeight = 4000; // mm
7
8      private static int passengerCarPrice = 100; // RUB
9      private static int cargoCarPrice = 250; // RUB
10     private static int vehicleAdditionalPrice = 200; // RUB
11
12     ▶ public static void main(String[] args) {
13         System.out.println("Сколько автомобилей сгенерировать?");
14
15         int carsCount = new Scanner(System.in).nextInt();
16
17         for (int i = 0; i < carsCount; i++) {
18             Car car = Camera.getNextCar();
19             System.out.println(car);
20
21             //Пропускаем автомобили спецтранспорта бесплатно
22             if (car.isSpecial) {
23                 System.out.println("Шлагбаум открывается... Счастливого пути!");
```

Это — код системы, которая контролирует проезд автомобилей в зависимости от разных параметров этих автомобилей.

Давайте разберём этот код.

Сначала идут параметры этой системы:

- `passengerCarMaxWeight=3500kg` — максимальный вес пассажирского автомобиля;
- `passengerCarMaxHeight=2000 mm` — максимальная высота пассажирского автомобиля;
- `controllerMaxHeight=4000 mm` — максимальная высота контроллера пропускного пункта;
- `passengerCarPrice=100 Rub` — стоимость проезда пассажирского автомобиля;
- `cargoCarPrice=250 Rub` — стоимость проезда грузового автомобиля;
- `vehicleAdditionalPrice=200 Rub` — дополнительная стоимость за проезд с прицепом.

Далее написан код. Сначала выводится в консоль вопрос «Сколько автомобилей сгенерировать?» и считывается введённое число из консоли.

Здесь работает цикл от 0 до введённого числа. Каждый раз здесь генерируется новый автомобиль, далее печатается информация об этом автомобиле в консоль, затем происходит проверка: если автомобиль является спецтранспортом, то в консоль выводится сообщение о том, что шлагбаум открывается и срабатывает оператор `continue`, то есть дальнейший код не выполняется и всё переходит к следующей итерации цикла.

Далее, если это условие не сработало, то код продолжает выполняться в текущей итерации цикла.

```

15     int carsCount = new Scanner(System.in).nextInt();
16
17     for (int i = 0; i < carsCount; i++) {
18         Car car = Camera.getNextCar();
19         System.out.println(car);
20
21         //Пропускаем автомобили спецтранспорта бесплатно
22         if (car.isSpecial) {
23             System.out.println("Шлагбаум открывается... Счастливого пути!");
24             continue;
25         }
26
27         //Проверяем высоту и массу автомобиля, вычисляем стоимость проезда
28         int price = calculatePrice(car);
29         if (price == -1) {
30             continue;
31         }
32
33         System.out.println("Общая сумма к оплате: " + price + " руб.");
34     }
35 }

```

Здесь мы получаем цену с помощью `calculatePrice` и, если цена равна (-1), то мы прерываем текущую итерацию цикла; если цена не равна (-1), то в консоль будет выведено сообщение «общая сумма к оплате» и указана сумма в рублях.

Теперь давайте разберёмся, что такое `calculatePrice`. Это ниже написанный код, который вычисляет стоимость проезда того или иного автомобиля в зависимости от написанных выше параметров.

```
38      * Расчёт стоимости проезда исходя из массы и высоты
39      */
40      @private static int calculatePrice(Car car) {
41          int carHeight = car.height;
42          int price = 0;
43          if (carHeight > controllerMaxHeight) {
44              System.out.println("Проезд невозможен: высота вашего " +
45                  "ТС превышает высоту пропускного пункта!");
46              return -1;
47          } else if (carHeight > passengerCarMaxHeight) {
48              double weight = car.weight;
49              //Грузовой автомобиль
50              if (weight > passengerCarMaxWeight) {
51                  price = passengerCarPrice;
52                  if (car.hasVehicle) {
53                      price = price + vehicleAdditionalPrice;
54                  }
55              }
56              //Легковой автомобиль
57              else {
58                  price = cargoCarPrice;
59              }
60          } else {
```

Давайте прочитаем, что здесь написано.

Сначала мы получаем высоту автомобиля, затем создаём переменную `price`, которая равна нулю; затем сравниваем: если высота автомобиля больше, чем высота пропускного пункта, то выводится сообщение о том, что проезд невозможен, поскольку высота вашего транспортного средства превышает высоту пропускного пункта, и возвращаем (-1). Разберём чуть позже, что означает этот возврат.

Далее, если это условие не выполняется, то происходит проверка второго условия — не превышает ли высота автомобиля максимально допустимую высоту пассажирского транспорта. Если превышает, то, в данном случае, берётся вес автомобиля и происходит проверка — больше ли этот вес, чем максимальный вес пассажирского автомобиля (3,5 тонны). Если этот вес больше, то устанавливается соответствующая цена и далее происходит проверка, есть ли у автомобиля прицеп. Если прицеп есть, то к цене

добавляется стоимость проезда с прицепом. В противном случае, если это условие не выполняется, то есть проезжает легковой автомобиль, то устанавливается другая цена и, если высота не превышает высоты пропускного пункта и не превышает максимальную высоту для пассажирского транспорта, то цена равна стоимости проезда пассажирского автомобиля. Таким образом происходит расчёт цены.

Этот фрагмент кода называется методом.

```
* Расчёт стоимости проезда исходя из массы и высоты  
*/  
private static int calculatePrice(Car car) {  
    int carHeight = car.height;  
    int price = 0;  
    if (carHeight > controllerMaxHeight) {  
        System.out.println("Проезд невозможен: высота вашего " +  
            "ТС превышает высоту пропускного пункта!");  
        return -1;  
    } else if (carHeight > passengerCarMaxHeight) {  
        double weight = car.weight;  
        //Грузовой автомобиль  
        if (weight > passengerCarMaxWeight) {  
            price = passengerCarPrice;  
            if (car.hasVehicle) {  
                price = price + vehicleAdditionalPrice;  
            }  
        }  
        //Легковой автомобиль  
    } else {  
        price = cargoCarPrice;  
    }  
} else {
```

Файлы с кодами, в которых расписана генерация новых автомобилей, некоторые свойства и вывод информации об автомобиле называются классами.

```
1 public class Camera {
2     @
3     public static Car getNextCar() {
4         String randomNumber = Double.toString(Math.random()).substring(2, 5);
5         int randomHeight = (int) (1000 + 3500. * Math.random());
6         double randomWeight = 600 + 10000 * Math.random();
7
8         Car car = new Car();
9         car.number = randomNumber;
10        car.height = randomHeight;
11        car.weight = randomWeight;
12        car.hasVehicle = Math.random() > 0.5;
13        car.isSpecial = Math.random() < 0.15;
14
15        return car;
16    }
17 }
```

Мы ещё не изучали классы, но, возможно, вам уже примерно понятно, как они работают. Мы изучим классы детально в следующих модулях.

#### **04:38—04:52**

Итак, в этом видеоматериале мы разобрали код достаточно сложного приложения, в котором используются различные изученные нами элементы синтаксиса языка Java.