**Лабораторная работа №1**

**Простейшие классы и объекты. Пакеты.**

Разработать классы для описанных ниже объектов. Включить в класс методы setX (…) – задает значение поля X объекта, getX(…) – возвращает значение поля X, show (…) – вывод на экран значений полей объекта. Определить для класса свои методы, позволяющие решить предложенные задачи.

Класс поместить в пакет. Обработать массив объектов, подключив к основной программе созданный пакет.

**9. Bus:** Фамилия и инициалы водителя, Номер автобуса, Номер маршрута, Марка, Год начала эксплуатации, Пробег.

Создать массив объектов. Вывести:

a) список автобусов для заданного номера маршрута;

b) список автобусов, которые эксплуатируются больше 10 лет;

c) список автобусов, пробег у которых больше 10000 км.

**Текст программы**

Файл **Bus.java**:

package Bus;  
  
public class Bus {  
 private String DriverSurname; *// ФИО водителя* private int BusNumber; *// Бортовой номер автобуса* private int RouteNumber; *// Номер маршрута* private String Mark; *// Марка автобуса* private int YearStart; *// Год начала эксплуатации* private int Mileage; *// Пробег* public Bus() {  
 DriverSurname = "No Driver";  
 BusNumber = 0;  
 RouteNumber = 0;  
 Mark = "No bus";  
 YearStart = 0;  
 Mileage = 0;  
 }  
  
 public Bus(String valueDriverSurname, int valueBusNumber, int valueRouteNumber, String valueMark, int valueYearStart, int valueMileage) {  
 DriverSurname = valueDriverSurname;  
 BusNumber = valueBusNumber;  
 RouteNumber = valueRouteNumber;  
 Mark = valueMark;  
 YearStart = valueYearStart;  
 Mileage = valueMileage;  
 }  
  
 public String getDriverSurname() {  
 return DriverSurname;  
 }  
  
 public void setDriverSurname(String driverSurname) {  
 DriverSurname = driverSurname;  
 }  
  
 public int getBusNumber() {  
 return BusNumber;  
 }  
  
 public void setBusNumber(int busNumber) {  
 BusNumber = busNumber;  
 }  
  
 public int getRouteNumber() {  
 return RouteNumber;  
 }  
  
 public void setRouteNumber(int routeNumber) {  
 RouteNumber = routeNumber;  
 }  
  
 public String getMark() {  
 return Mark;  
 }  
  
 public void setMark(String mark) {  
 Mark = mark;  
 }  
  
 public int getYearStart() {  
 return YearStart;  
 }  
  
 public void setYearStart(int yearStart) {  
 YearStart = yearStart;  
 }  
  
 public int getMileage() {  
 return Mileage;  
 }  
  
 public void setMileage(int mileage) {  
 Mileage = mileage;  
 }  
  
 public void show() {  
 System.*out*.println("<-------------------->" +  
 "\nВодитель: " + DriverSurname +  
 "\nБортовой номер автобуса: " + BusNumber +  
 "\nНомер маршрута: " + RouteNumber +  
 "\nМарка автобуса: " + Mark +  
 "\nГод начала эксплуатации: " + YearStart +  
 "\nПробег: " + Mileage +  
 "\n<-------------------->");  
 }  
  
 public int checkRoute(int route, int count) {  
 if (this.RouteNumber == route & count == 0) {  
 System.*out*.print("Список автобусов, работающих по " + route + " маршруту:\n");  
 this.show();  
 count++;  
 } else if (this.RouteNumber == route & count != 0) {  
 this.show();  
 count++;  
 }  
 return count;  
 }  
  
 public int checkYear(int count) {  
 if (2025 - this.YearStart > 10 & count == 0) {  
 System.*out*.print("Список автобусов, которые эксплуатируются более 10 лет:\n");  
 this.show();  
 count++;  
 } else if (2025 - this.YearStart > 10 & count != 0) {  
 this.show();  
 count++;  
 }  
 return count;  
 }  
  
 public int checkMileage(int count) {  
 if (this.Mileage > 10000 & count == 0) {  
 System.*out*.print("Список автобусов с пробегом более 10000 км:\n");  
 this.show();  
 count++;  
 } else if (this.Mileage > 10000 & count != 0) {  
 this.show();  
 count++;  
 }  
 return count;  
 }  
}

Файл **Main.java**:

import Bus.Bus;  
import java.util.ArrayList;  
import java.util.Scanner;  
  
public class Main {  
 public static void main(String[] args) {  
 ArrayList<Bus> Buses = new ArrayList<>();  
 Bus bus1 = new Bus("Маркелов", 130, 2, "НеФАЗ-5299", 2024, 1000);  
 Bus bus2 = new Bus("Подтелков", 122, 2, "НеФАЗ-5299", 2024, 1500);  
 Bus bus3 = new Bus("Фатькина", 542, 8, "ЛиАЗ-5292", 2021, 4000);  
 Bus bus4 = new Bus("Беляков", 1308, 18, "МАЗ-103", 2017, 15000);  
 Bus bus5 = new Bus("Бурмистров", 509, 12, "ЛиАЗ-5292", 2019, 12000);  
 Bus bus6 = new Bus("Минеев", 109, 9, "НеФАЗ-5299", 2024, 2000);  
 Bus bus7 = new Bus("Цветков", 717, 7, "МАЗ-103", 2013, 24000);  
  
 Buses.add(bus1);  
 Buses.add(bus2);  
 Buses.add(bus3);  
 Buses.add(bus4);  
 Buses.add(bus5);  
 Buses.add(bus6);  
 Buses.add(bus7);  
  
 System.*out*.println(  
 "1. Вывести список автобусов по номеру маршрута\n" +  
 "2. Вывести список автобусов, которые эксплуатируются более 10 лет\n" +  
 "3. Вывести список автобусов с пробегом более 10000 км\n" +  
 "Введите номер операции: ");  
 Scanner in = new Scanner(System.*in*);  
 int n = in.nextInt();  
 switch (n) {  
 case 1: {  
 System.*out*.println("Введите номер маршрута: ");  
 Scanner in1 = new Scanner(System.*in*);  
 int route = in.nextInt();  
 int count = 0;  
 for (Bus b : Buses) count = b.checkRoute(route, count);  
 if (count == 0) System.*out*.println("За данным маршрутом не закреплены автобусы, либо такого маршрута не существует");  
 in1.close();  
 break;  
 }  
 case 2: {  
 int count = 0;  
 for (Bus b : Buses) count = b.checkYear(count);  
 if (count == 0) System.*out*.println("Нет автобусов, эксплуатирующихся более 10 лет");  
 break;  
 }  
 case 3: {  
 int count = 0;  
 for (Bus b : Buses) count = b.checkMileage(count);  
 if (count == 0) System.*out*.println("Нет автобусов с пробегом более 10000 км");  
 break;  
 }  
 default: System.*out*.println("Такой операции нет!");  
 }  
 in.close();  
 }  
}

**Описание программы**

Класс **Bus** содержит поля:

* **DriverSurname** – фамилия водителя
* **BusNumber** – бортовой номер автобуса
* **RouteNumber** – номер маршрута
* **Mark** – марка автобуса
* **YearStart** – год начала эксплуатации автобуса
* **Mileage** – пробег автобуса

Класс **Bus** содержит методы:

* **Bus()** – конструктор, инициализирующий объекты класса Bus
* **getDriverSurname** – геттер для фамилии водителя
* **getBusNumber** – геттер для бортового номера автобуса
* **getRouteNumber** – геттер для номера маршрута
* **getMark** – геттер для марки автобуса
* **getYearStart** – геттер для года начала эксплуатации автобуса
* **getMileage** – геттер для пробега автобуса
* **setDriverSurname** – сеттер для фамилии водителя
* **setBusNumber** – сеттер для бортового номера автобуса
* **setRouteNumber** –сеттер для номера маршрута
* **setMark** – сеттер для марки автобуса
* **setYearStart** – сеттер для года начала эксплуатации автобуса
* **setMileage** – сеттер для пробега автобуса
* **show()** – функция, выводящая значения полей объекта
* **checkRoute()** – функция, сравнивающая номер маршрута с введенным пользоватем:
  + На вход подается введенный пользователем номер маршрута **route** и счетчик автобусов, закрепленных за ним **count** (изначально равен 0)
  + Если номер текущего маршрута совпадает с маршрутом **route**, а **count** равен 0, выводится сообщение «*Список автобусов, работающих по* ***\*route\**** *маршруту»*. Затем вызывается функция **show** (выводится информация о данном автобусе) и значение **count** увеличивается на 1
  + Если номер текущего маршрута совпадает с маршрутом **route**, а **count** не равен 0, вызывается функция **show** (выводится информация о данном автобусе) и значение **count** увеличивается на 1
  + Функция возвращает значение **count**
* **checkYear()** – функция, определяющая автобусы, эксплуатирующиеся более 10 лет:
  + На вход подается счетчик автобусов, эксплуатирующихся более 10 лет, **count** (изначально равен 0)
  + Если значение выражения 2025 – **this.YearStart** (2025 – год выпуска автобуса) более 10, а **count** равен 0, выводится сообщение «*Список автобусов, которые эксплуатируются более 10 лет»*. Затем вызывается функция **show** (выводится информация о данном автобусе) и значение **count** увеличивается на 1
  + Если значение выражения 2025 – **this.YearStart** (2025 – год выпуска автобуса) более 10, а **count** не равен 0, вызывается функция **show** (выводится информация о данном автобусе) и значение **count** увеличивается на 1
  + Функция возвращает значение **count**
* **checkMileage()** – функция, определяющая автобусы с пробегом более 10000 км:
  + На вход подается счетчик автобусов с пробегом более 10000 км **count** (изначально равен 0)
  + Если значение пробега **this.Mileage** более 10000, а **count** равен 0, выводится сообщение «*Список автобусов с пробегом более 10000 км»*. Затем вызывается функция **show** (выводится информация о данном автобусе) и значение **count** увеличивается на 1
  + Если значение выражения пробега **this.Mileage** более 10000, а **count** не равен 0, вызывается функция **show** (выводится информация о данном автобусе) и значение **count** увеличивается на 1
  + Функция возвращает значение **count**

Функция **Main**:

1. Инициализируется массив **Buses**, содержащий объекты типа Bus
2. Инициализируются объекты типа Bus (автобусы) – **bus1**, **bus2**, **bus3**, **bus4**, **bus5**, **bus6**, **bus7**
3. Данные объекты добавляются в массив Buses
4. Программа выводит сообщение, предлагающее выбрать одну из трех операций:
5. Вывести список автобусов по номеру маршрута
6. Вывести список автобусов, которые эксплуатируются более 10 лет
7. Вывести список автобусов с пробегом более 10000 км
8. Программа запрашивает номер операции **n**
9. В зависимости от введенного номера программа производит следующие действия:
10. При **n = 1**:

* Программа запрашивает номер маршрута **route**
* Счетчику автобусов **count** присваивается значение 0
* С помощью цикла **for** перебираются все автобусы из массива **Buses**. Для каждого автобуса вызывается функция **checkRoute()**, которая выводит информацию об автобусе и увеличивает **count**, если номер маршрута совпадает с **route**
* Если после полного прохождения цикла переменная count останется равной 0, выводится сообщение *«За данным маршрутом не закреплены автобусы, либо такого маршрута не существует»*

1. При **n = 2**:

* Счетчику автобусов **count** присваивается значение 0
* С помощью цикла **for** перебираются все автобусы из массива **Buses**. Для каждого автобуса вызывается функция **checkYear()**, которая выводит информацию об автобусе и увеличивает **count**, если значение поля **Year** у данного автобуса более 10
* Если после полного прохождения цикла переменная count останется равной 0, выводится сообщение *«Нет автобусов, эксплуатирующихся более 10 лет»*

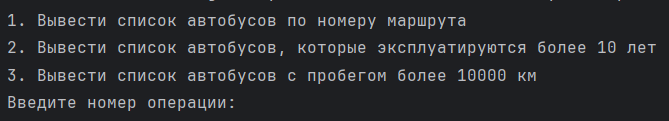
1. При **n = 3**:

* Счетчику автобусов **count** присваивается значение 0
* С помощью цикла **for** перебираются все автобусы из массива **Buses**. Для каждого автобуса вызывается функция **checkMileage()**, которая выводит информацию об автобусе и увеличивает **count**, если значение поля **Mileage** у данного автобуса более 10000
* Если после полного прохождения цикла переменная count останется равной 0, выводится сообщение *«Нет автобусов с пробегом более 10000 км»*

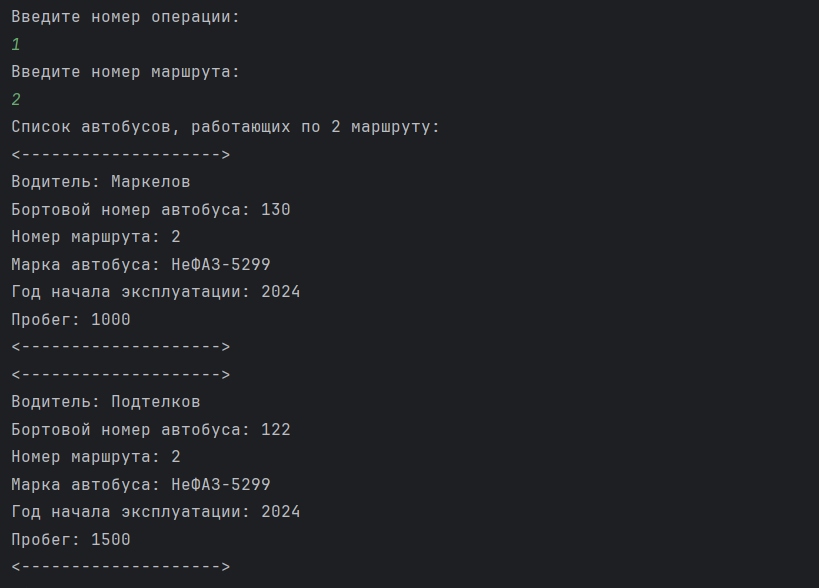
1. При любом другом значении **n**:
   * Выводится сообщение *«Такой операции нет!»*

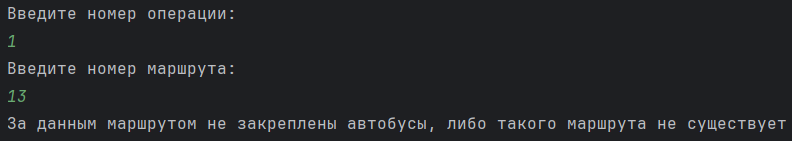
**Результаты тестов**

Стартовое сообщение:

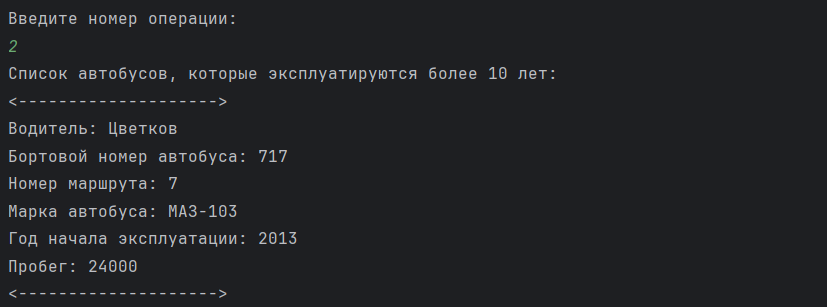


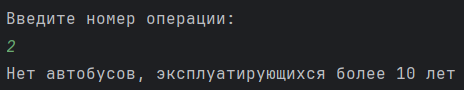
Вывод автобусов по номеру маршрута:



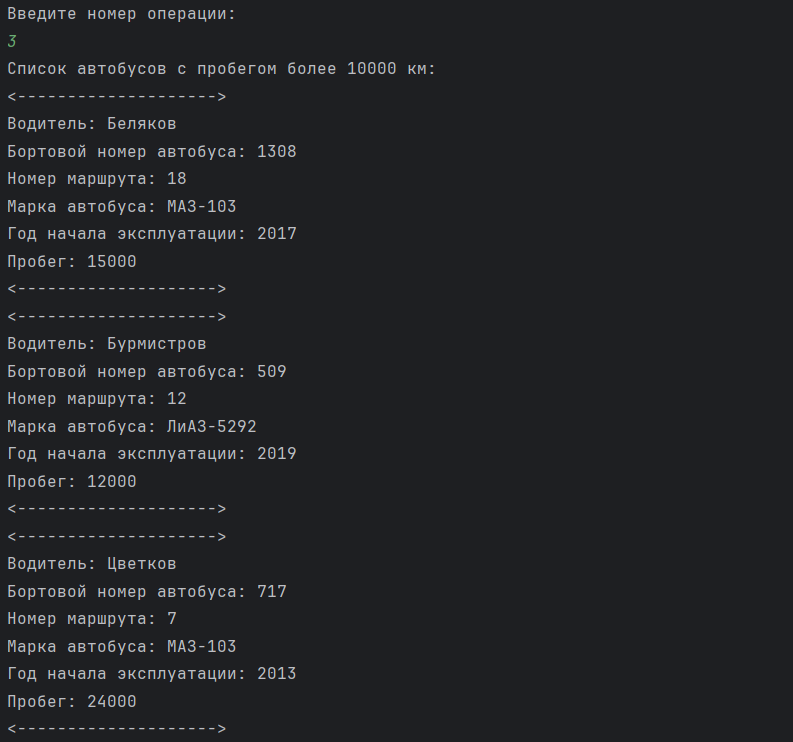


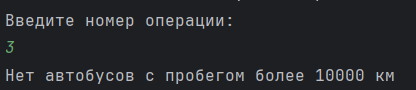
Вывод автобусов, эксплуатирующихся более 10 лет:



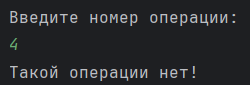


Вывод автобусов с пробегом более 10000 км:





Ввод несуществующей операции:



**Вывод**

В ходе выполнения лабораторной работы был разработан класс **Bus**, описывающий автобусы и содержащий поля: фамилия водителя, бортовой номер автобуса, номер маршрута, марка автобуса, год начала эксплуатации и пробег. В класс были добавлены сеттеры (функции, задающие значения полей объекта), геттеры (функции, возвращающие значения полей объекта) и функция **show** (выводит на экран значения полей объекта). Также были добавлены 3 метода для решения задач лабораторной работы:

* Метод, выводящий автобусы по выбранному пользователем маршруту
* Метод, выводящий автобусы, эксплуатирующиеся старше 10 лет
* Метод, выводящий автобусы с пробегом более 10000 км

Объекты типа **Bus** были помещены в массив объектов **Buses**