Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «Практическая работа 9.2 Kotlin»

Выполнил: Идрисов Руслан

Группа: ПР-22

Преподаватель: Мирошниченко Г.В

2023

**Задание: 1.** Задача на взаимодействие между классами. Разработать систему «Автобаза». Диспетчер распределяет заявки на Рейсы между Водителями и назначает для этого Автомобиль. Водитель может сделать заявку на ремонт. Диспетчер может отстранить Водителя от работы. Водитель делает отметку о выполнении Рейса и состоянии Автомобиля.

**Входные данные:**

**Класс Car:** Number (string) – номер автомобиля, Sostoyanie (string) – состояние автомобиля.

**Класс Voditel:** Name (string) – имя водителя

**Класс Dispetcher:** voditel (Voditel), car (Car)

**Выходные данные**

var Name = *readln*() ?: ""

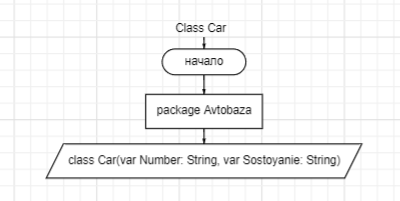
var Number = *readln*() ?: ""

voditeli.*forEach***{** *println*(**it**.Name) **}**

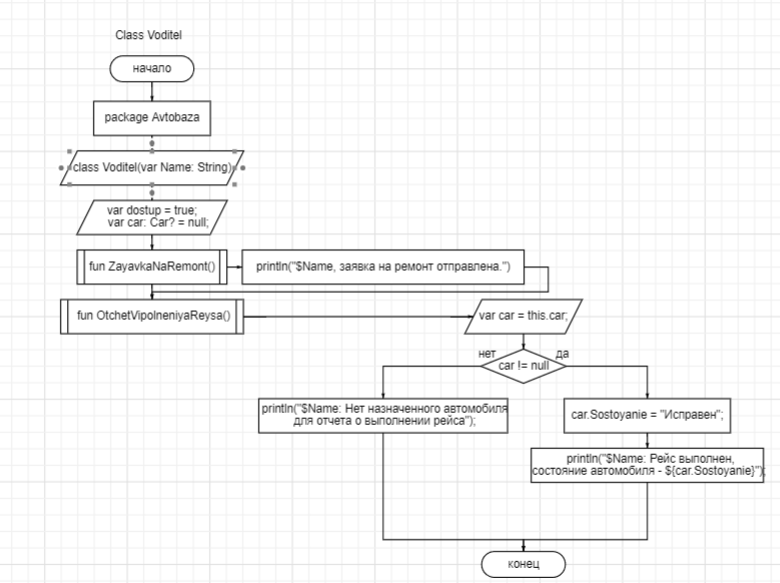
cars.*forEach***{** (*println*(**it**.Number)) **}**

**Блок-схема**

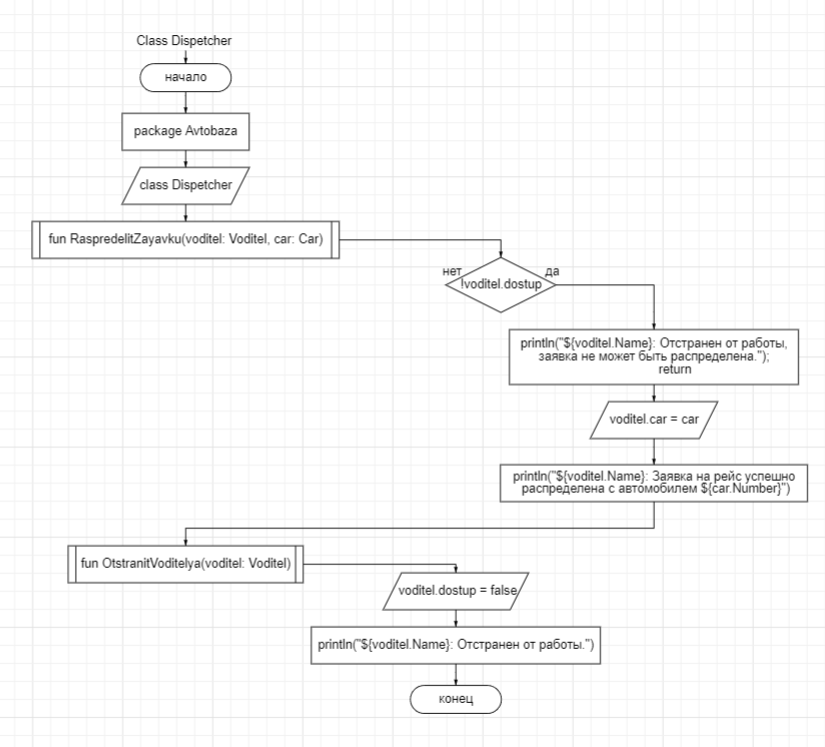
**Car:**

****

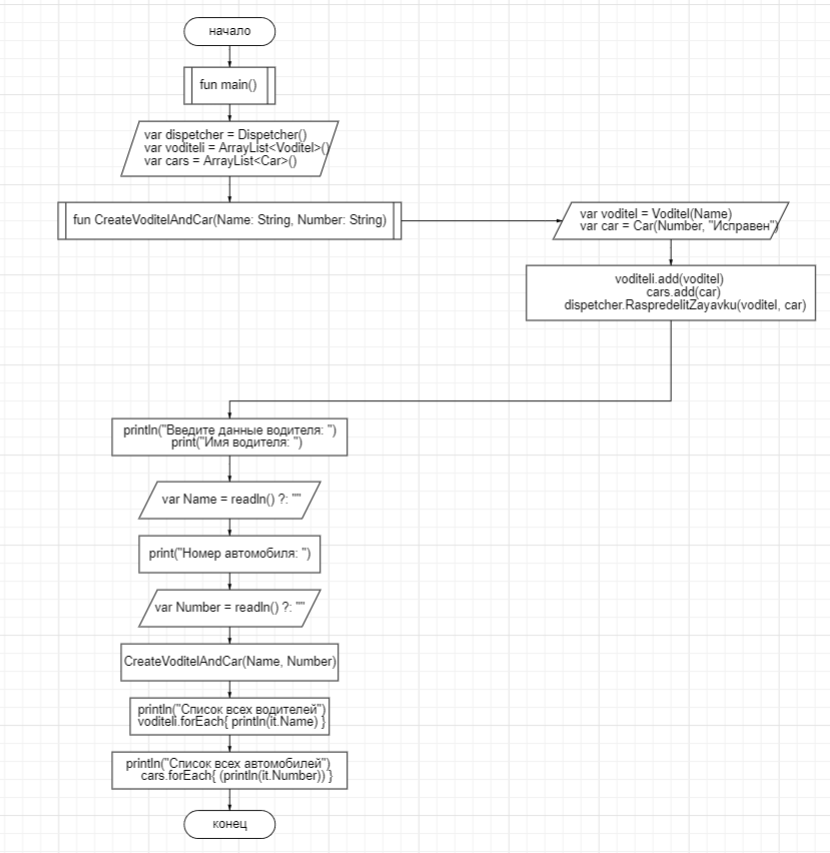
**Voditel:**

****

**Dispetcher:**

****

**Main**

****

**Листинг программы (если есть)**

**Класс Car:**

class Car(var Number: String, var Sostoyanie: String) {  
  
}

**Класс Voditel:**

package Avtobaza  
class Voditel(var Name: String) {  
 var dostup = true;  
 var car: Car? = null;  
 fun ZayavkaNaRemont()  
 {  
  
 *println*("$Name, заявка на ремонт отправлена.")  
 }  
 fun OtchetVipolneniyaReysa()  
 {  
 var car = this.car;  
 if (car != null)  
 {  
 car.Sostoyanie = "Исправен";  
 *println*("$Name: Рейс выполнен, состояние автомобиля - ${car.Sostoyanie}");  
  
 }  
 else  
 {  
 *println*("$Name: Нет назначенного автомобиля для отчета о выполнении рейса");  
 }  
 }  
}

**Класс Dispetcher:**

package Avtobaza  
class Dispetcher {  
 fun RaspredelitZayavku(voditel: Voditel, car: Car)  
 {  
 if (!voditel.dostup)  
 {  
 *println*("${voditel.Name}: Отстранен от работы, заявка не может быть распределена.");  
 return  
 }  
 voditel.car = car  
 *println*("${voditel.Name}: Заявка на рейс успешно распределена с автомобилем ${car.Number}")  
  
 }  
 fun OtstranitVoditelya(voditel: Voditel)  
 {  
 voditel.dostup = false  
 *println*("${voditel.Name}: Отстранен от работы.")  
 }  
}

**Main:**

package Avtobaza  
fun main() {  
 var dispetcher = Dispetcher()  
 var voditeli = ArrayList<Voditel>()  
 var cars = ArrayList<Car>()  
  
 fun CreateVoditelAndCar(Name: String, Number: String)  
 {  
 var voditel = Voditel(Name)  
 var car = Car(Number, "Исправен")  
 voditeli.add(voditel)  
 cars.add(car)  
 dispetcher.RaspredelitZayavku(voditel, car)  
 }  
  
 *println*("Введите данные водителя: ")  
 *print*("Имя водителя: ")  
 var Name = *readln*() ?: ""  
 *print*("Номер автомобиля: ")  
 var Number = *readln*() ?: ""  
  
 CreateVoditelAndCar(Name, Number)  
  
 *println*("Список всех водителей")  
 voditeli.*forEach***{** *println*(**it**.Name) **}** *println*("Список всех автомобилей")  
 cars.*forEach***{** (*println*(**it**.Number)) **}**}