Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практическое задание 9А(1)**»

Выполнил: Епимахов Денис Леонидович

Группа: ПР-21

Преподаватель: Мирошниченко Г. В.

2025

**Вариант 14**

**Задание:**



**Входные и выходные данные:**

Car – базовый класс; class

brand – марка, поле класса Car; строка, string

power - мощность, поле класса Car; вещественное число, double

price – цена, поле класса Car; вещественное число, double

petrol – булева переменная (бензин или дизель), поле класса Car; булева переменная, Boolean

probeg - пробег поле класса Car; целое число, int

InfoCar() – информация о машине, функция класса Car; fun

Power() – информация о мощности машины, функция класса Car; fun

ToplivoCar() – информация о топливе машины, функция класса Car; fun

Probeg() – информация о пробеге машины, функция класса Car; fun

Track – дочерний класс Car; class

brand – марка, поле класса Track; строка, string

power - мощность, поле класса Track; вещественное число, double

price – цена, поле класса Track; вещественное число, double

liftingCapacity – грузоподъемность, поле класса Track; вещественное число, double

petrol – булева переменная (бензин или дизель), поле класса Track; булева переменная, Boolean

probeg - пробег поле класса Track; целое число, int

InfoTrack() – информация о грузовике, функция класса Track; fun

LiftingCapacityTrack() – информация о грузоподъёмности грузовика, функция класса Track; fun

ToplivoTrack() – информация о топливе грузовика, функция класса Track; fun

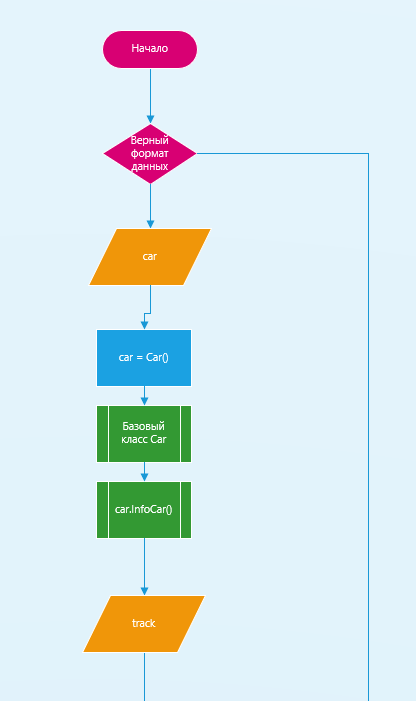
CalculateFuelConsumption(distance – пройденная дистанция; вещественное число, double; fuelUsed – потраченное топливо; вещественное число, double) – информация о топливе грузовика, функция класса Track; fun

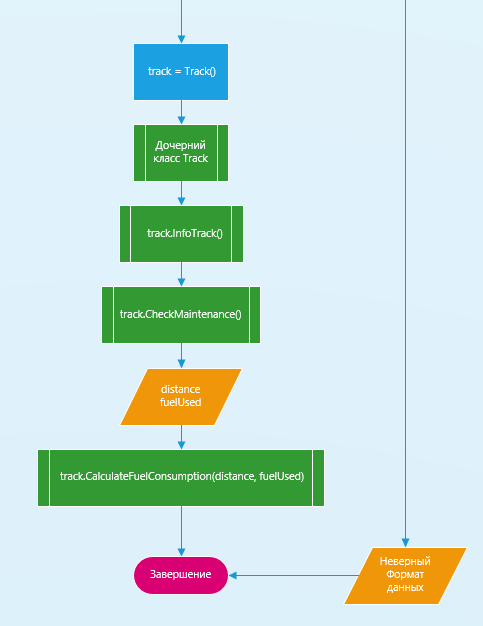
ToplivoTrack() – информация о топливе грузовика, функция класса Track; fun

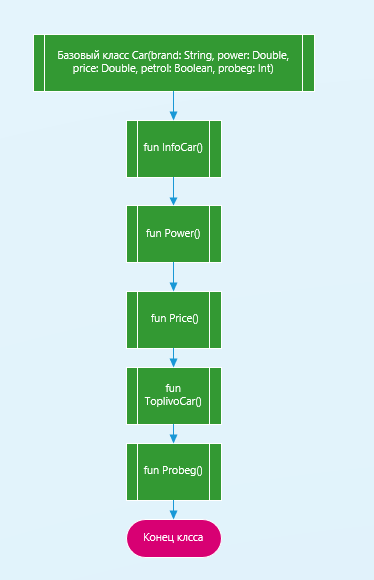
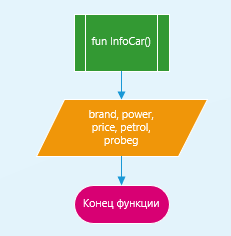
CheckMaintenance() – информация проверки необходимости технического обслуживания, функция класса Track; fun

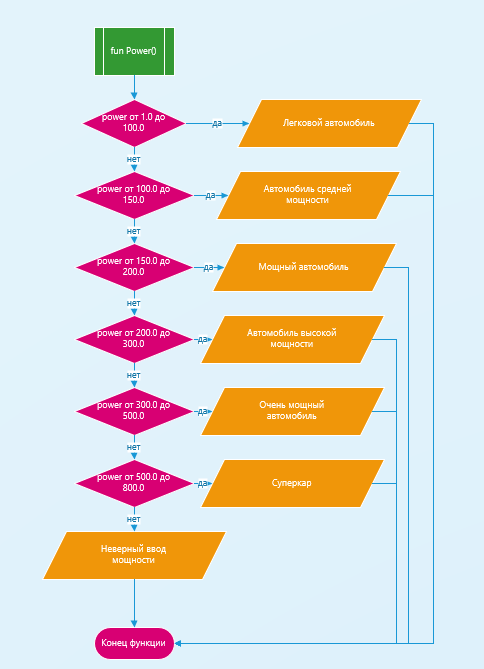
PowerTrack() – информация о мощности грузовика, функция класса Track; fun

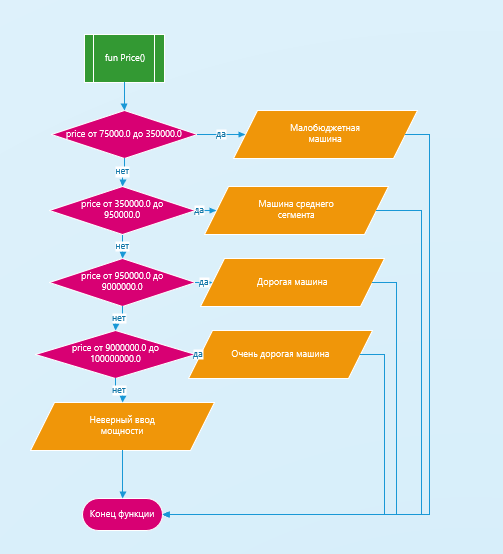
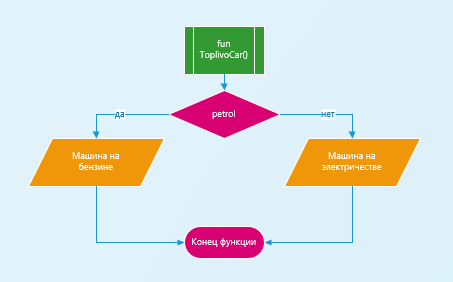
**Блок-схема:**

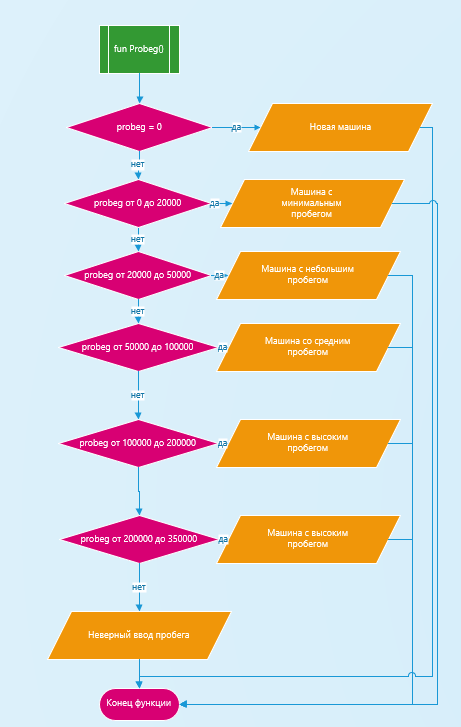
****

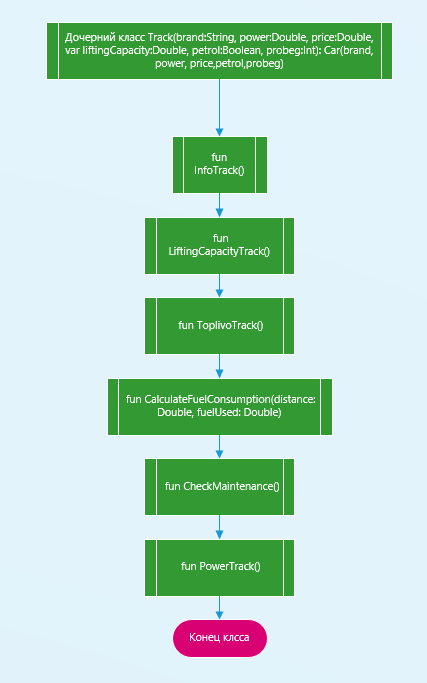
****

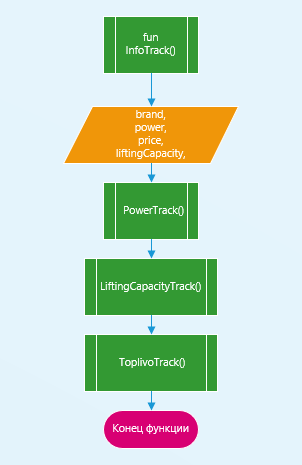
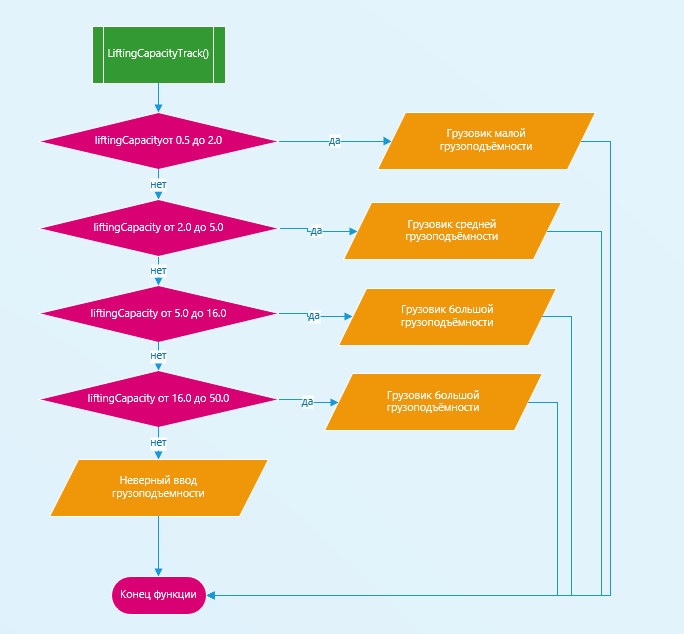
****

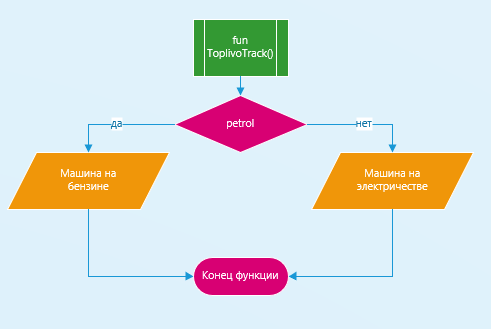
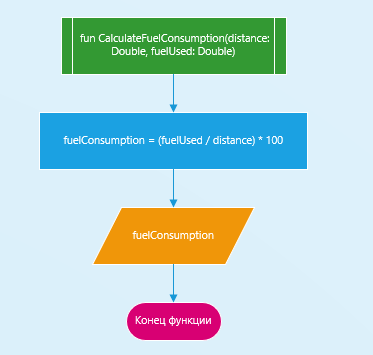
****

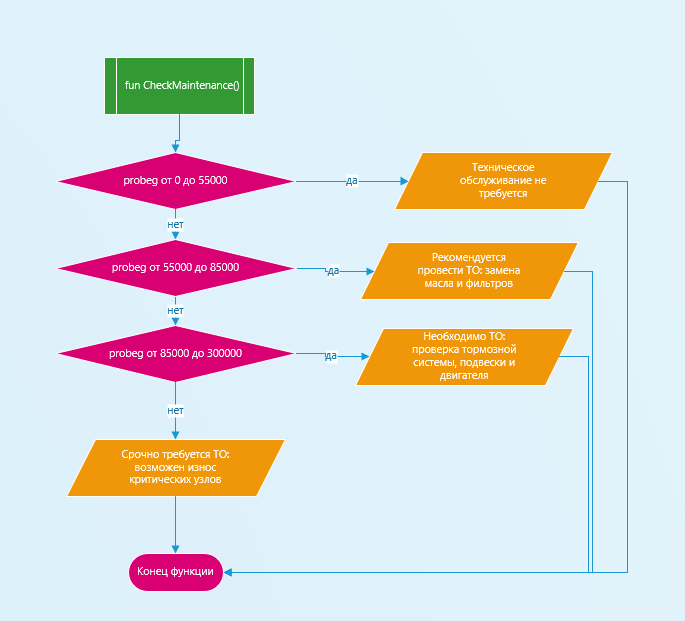
****

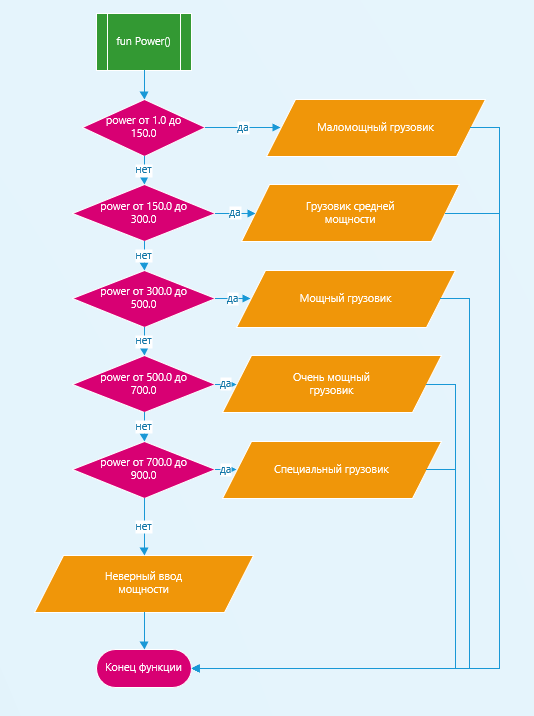
****

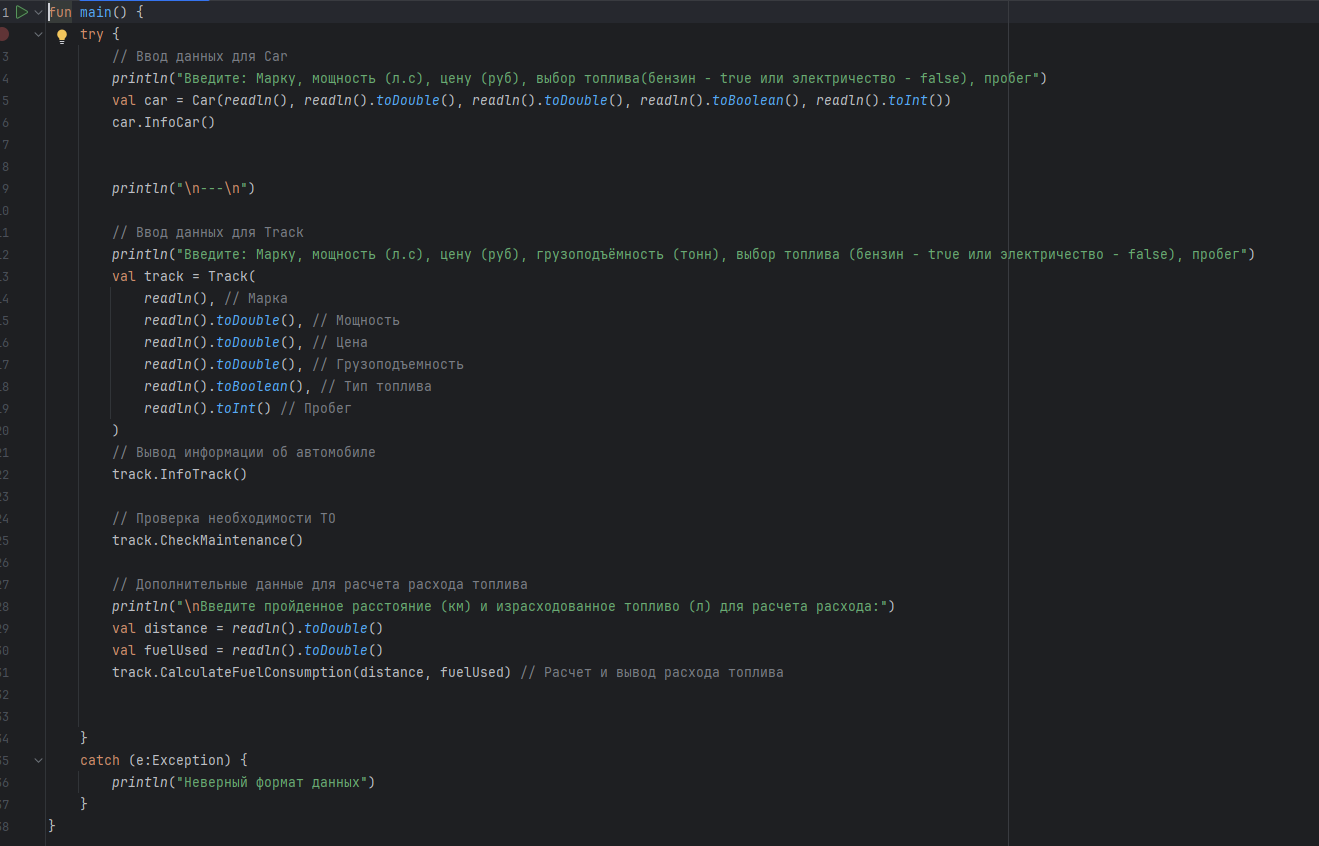
****

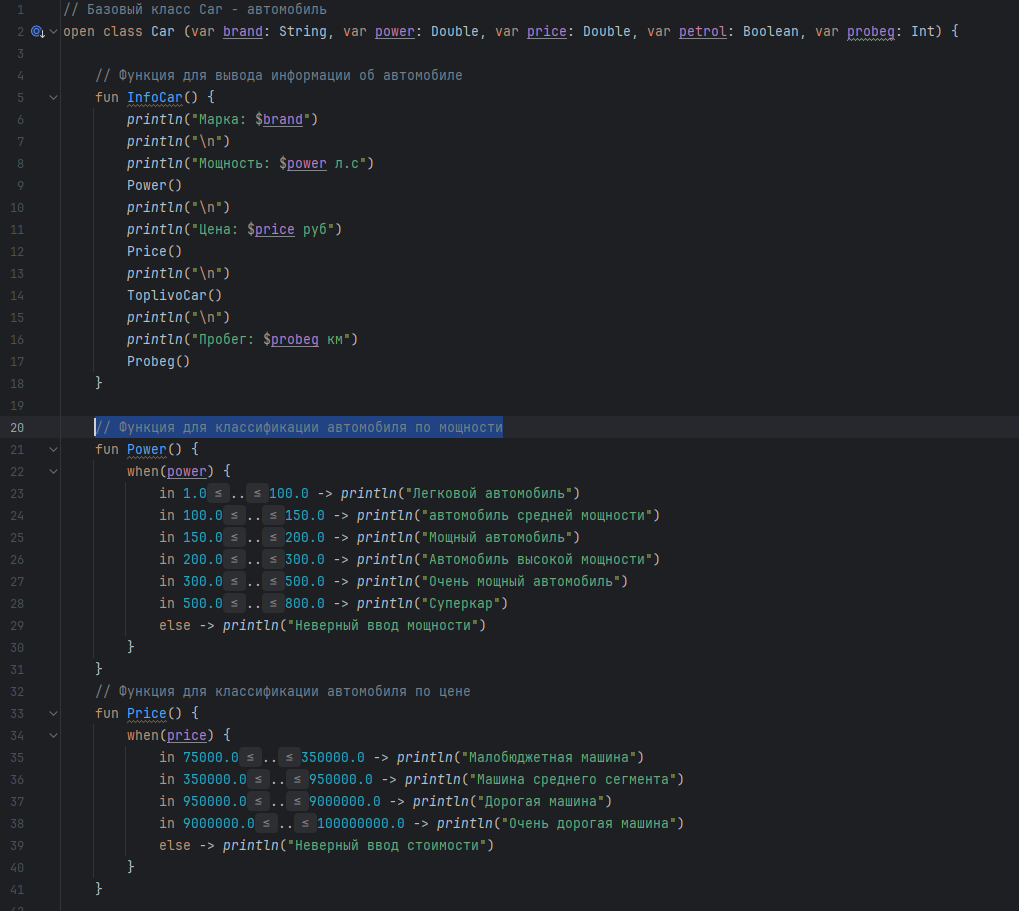
****

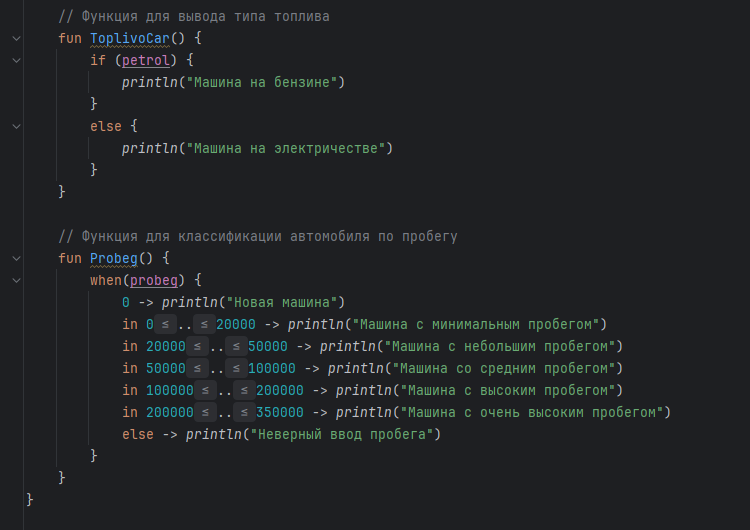
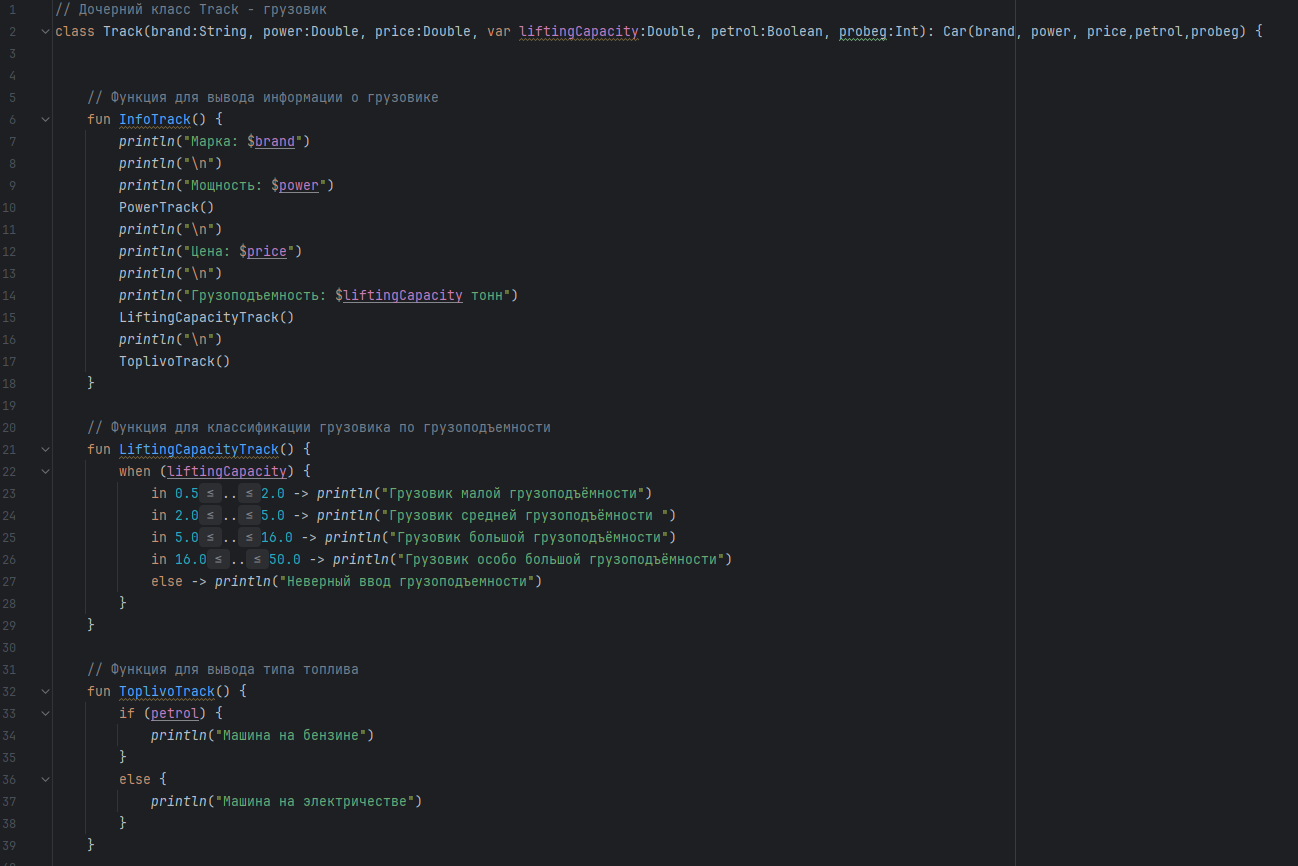
****

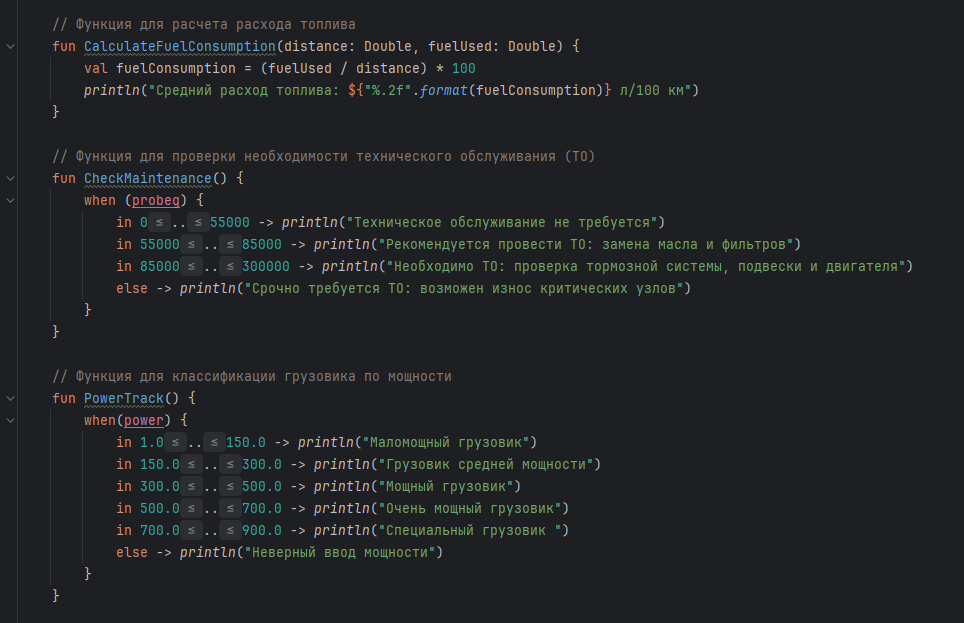
****

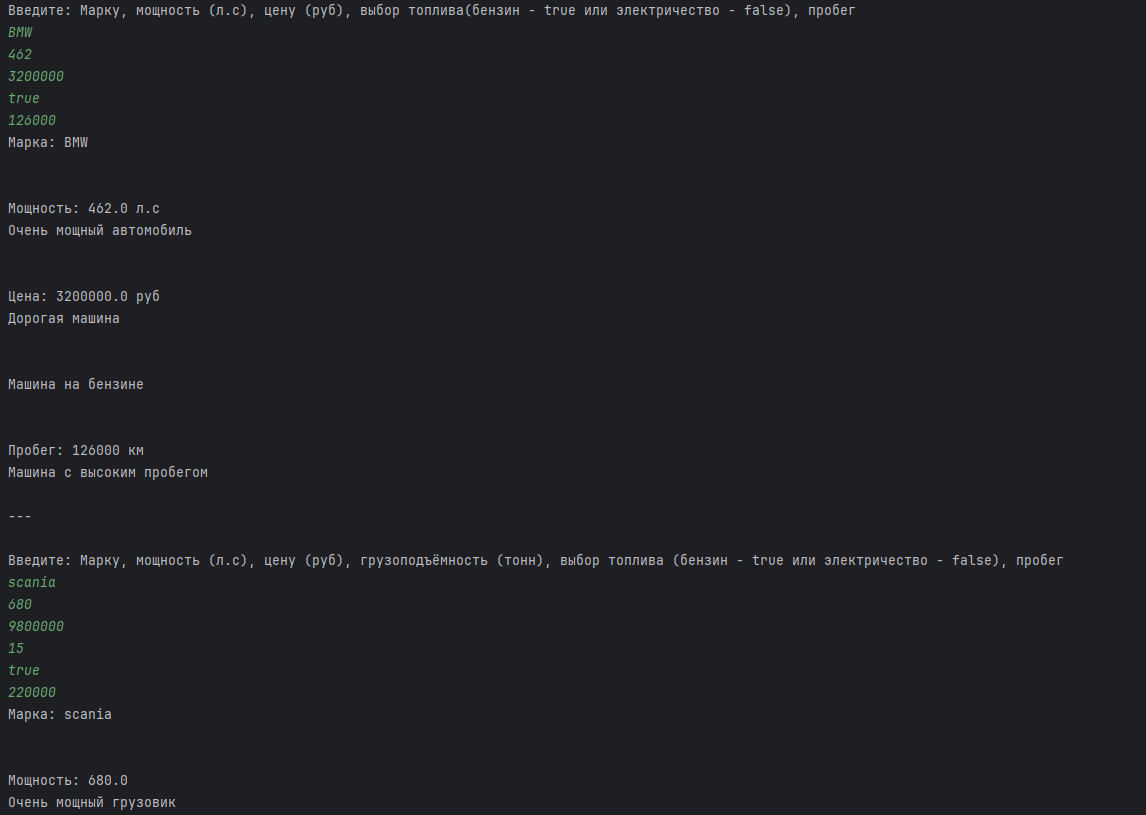
****

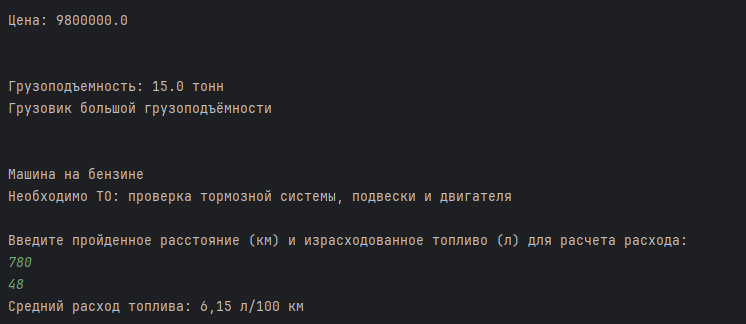
**Листинг программы:**

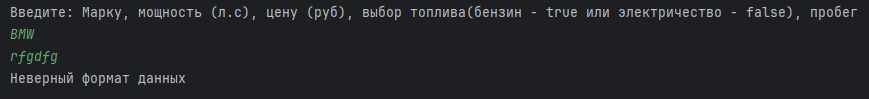
****

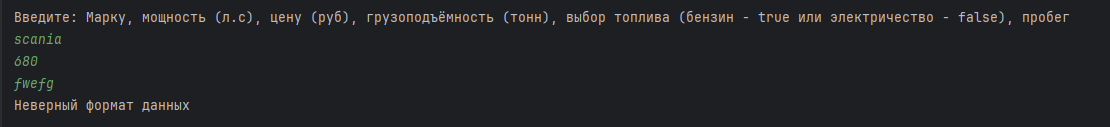
****

****

**Тестовые ситуации:**

****

****

****

**Вариант 10**

**Задание:**



**Входные и выходные данные:**

PhoneCall – базовый класс; class

phoneNumber – номер телефона, поле класса PhoneCall; строка, string

callDate - дата разговора (в формате "дд.мм.гггг"), поле класса PhoneCall; строка, string

duration – продолжительность звонка в минутах, поле класса PhoneCall; целое число, int

cityCode – булева переменная (бензин или дизель), поле класса PhoneCall; целое число, int

callPricePerMinute - цена разговора в минуту, поле класса PhoneCall; вещественное число, double

PhoneCallInfo() – информация о звонке, функция класса PhoneCall; fun

CalculateCallCost() – информация о стоимости звонка, функция класса PhoneCall; вещественное число, double, fun

GetDayOfWeek() – определение дня недели звонка, функция класса PhoneCall; строка, string, fun

IsLuckyCityCode() – является ли код города "счастливым", функция класса PhoneCall; fun

RoamingCall– дочерний класс PhoneCall; class

phoneNumber – номер телефона, поле класса PhoneCall; строка, string

callDate - дата разговора (в формате "дд.мм.гггг"), поле класса PhoneCall; строка, string

duration – продолжительность звонка в минутах, поле класса PhoneCall; целое число, int

cityCode – булева переменная (бензин или дизель), поле класса PhoneCall; целое число, int

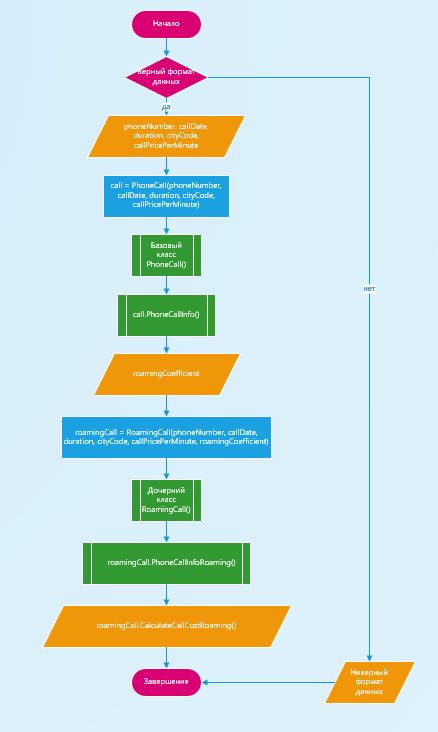
callPricePerMinute - цена разговора в минуту, поле класса PhoneCall; вещественное число, double

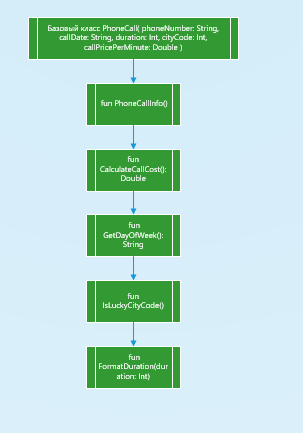
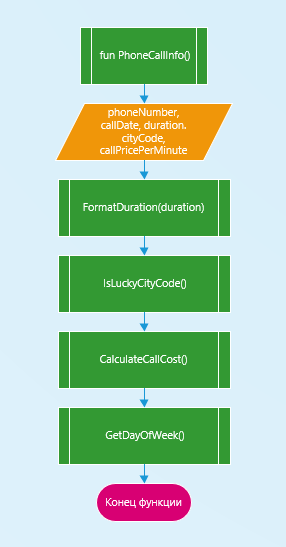
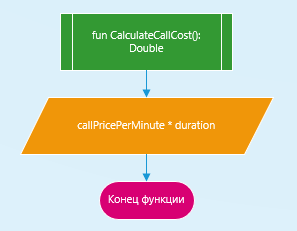
roamingCoefficient- коэффициент увеличения стоимости в роуминге, поле класса PhoneCall; вещественное число, double

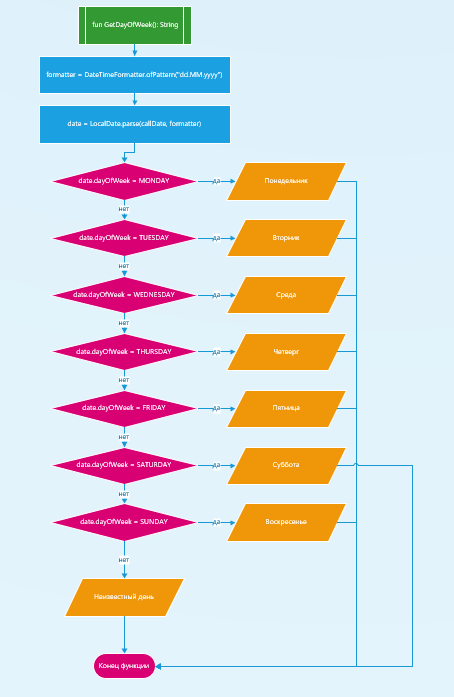
CalculateCallCostRoaming() – информация о стоимости звонка с учетом роуминга, функция класса RoamingCall; вещественное число, double, fun

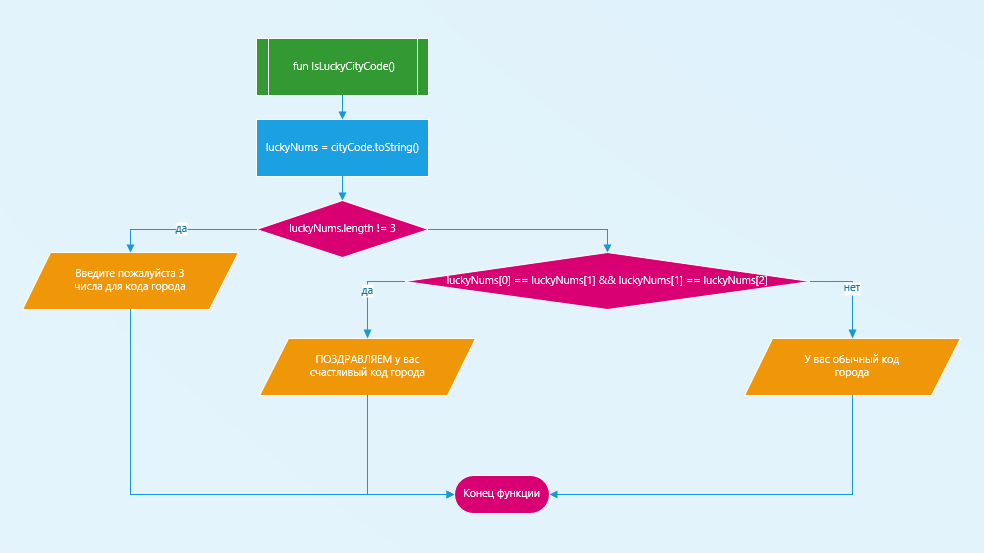
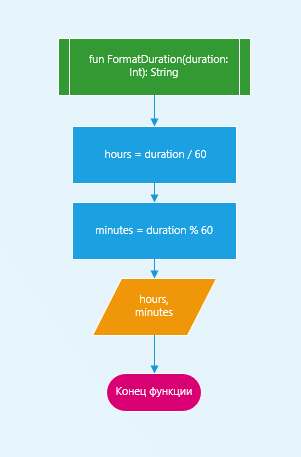
PhoneCallInfoRoaming() – информация о звонке с учетом роуминга, функция класса RoamingCall; fun

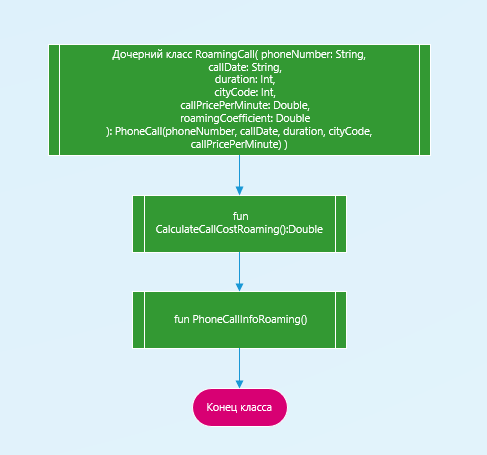
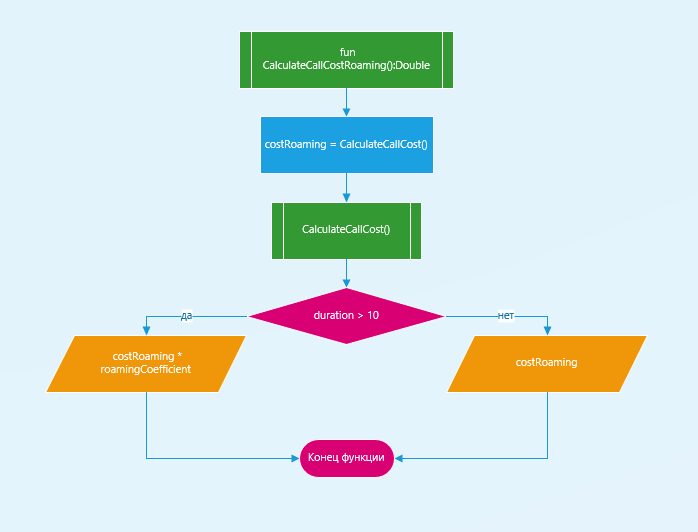
**Блок-схема:**

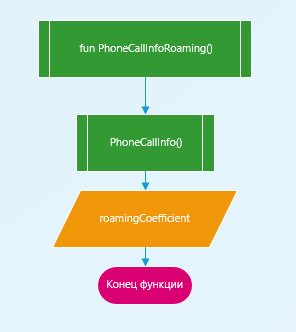


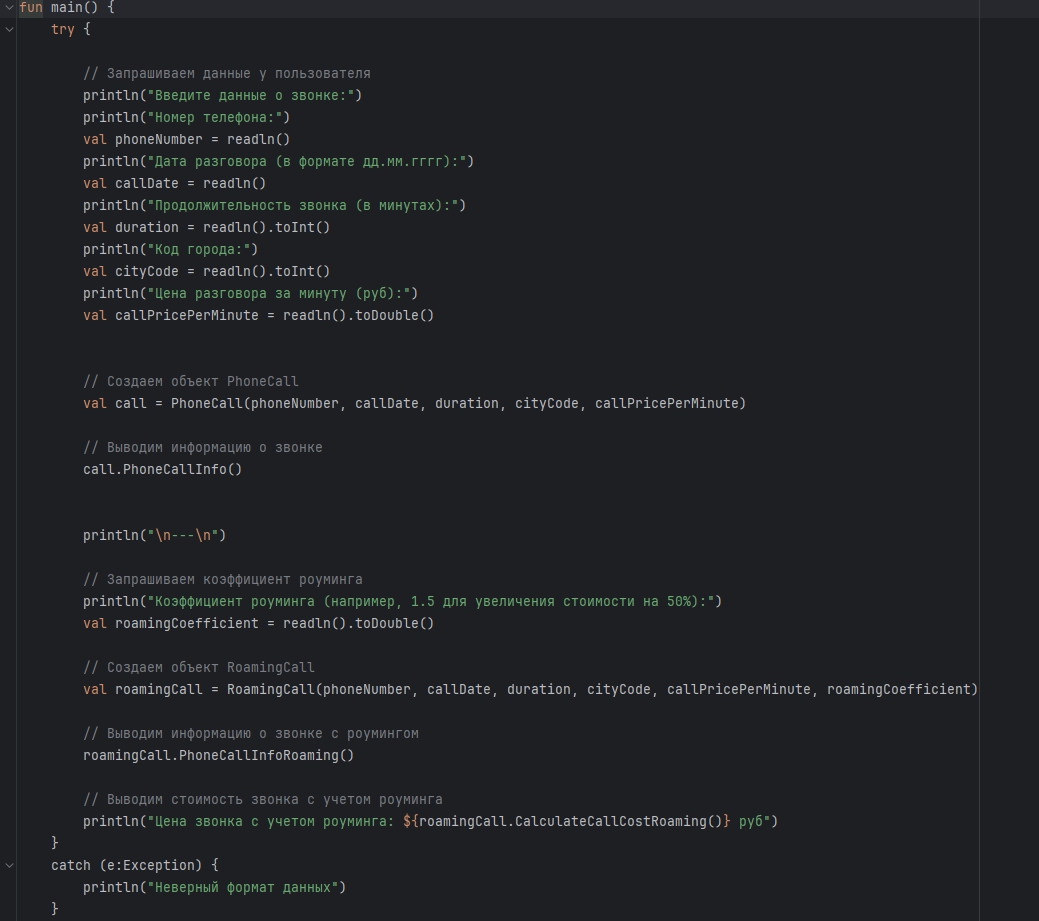
****

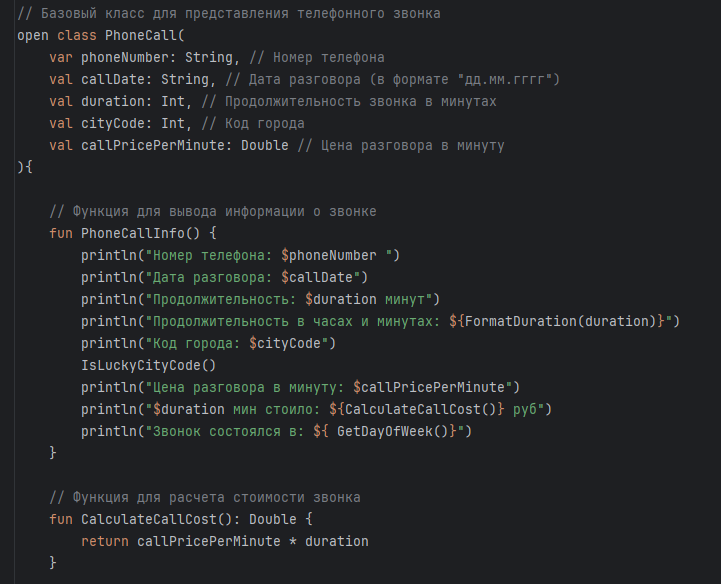
****

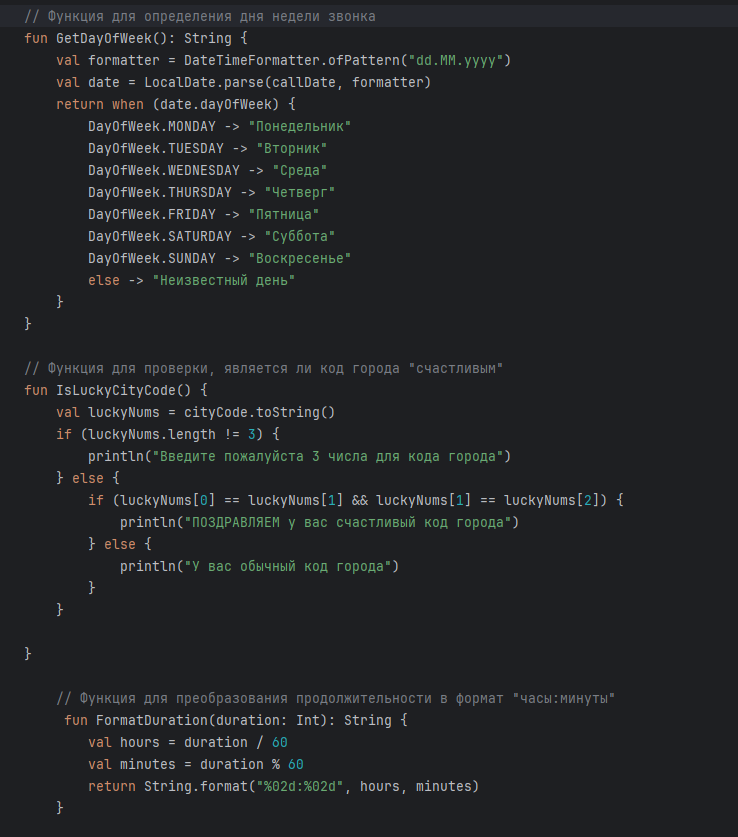
****

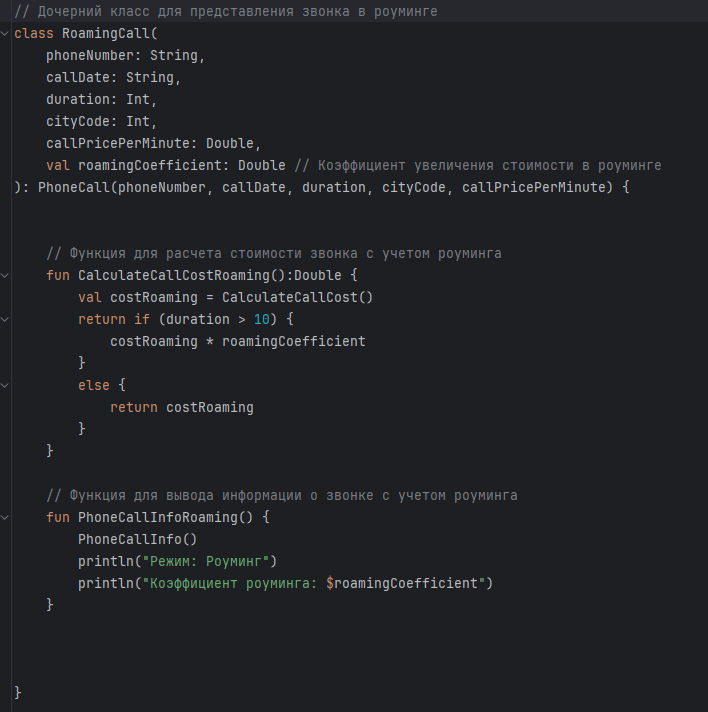
****

****

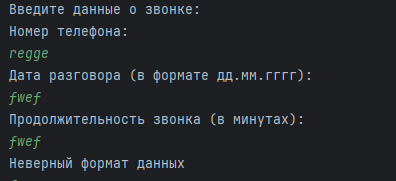
**Листинг программы:**

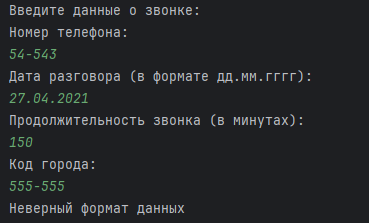
****

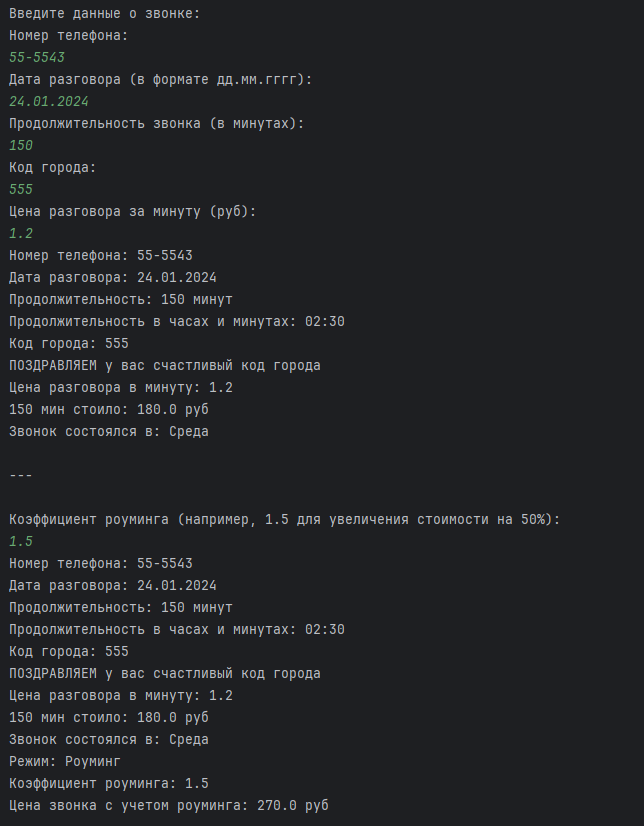
****

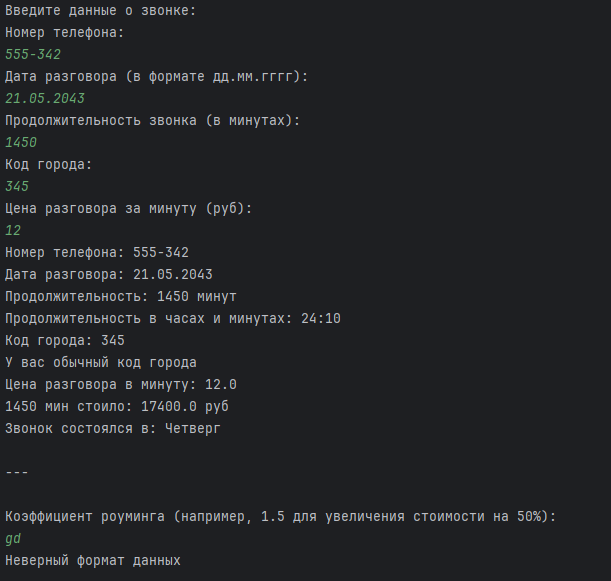
****

**Тестовые ситуации:**



****



****

**Вывод:** я выполнил задание в Kotlin и создал блок-схемы.