Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «Пакеты и импорт, наследование, модификаторы видимости в Kotlin»

Выполнил: Шурков Дмитрий Александрович

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2023

**Задание 2:** Класс «Насекомые»: название, скорость передвежения, окраска, используемая пища, среда обитания.

**Входные данные:**

name – Название насекомого (String)

**Выходные данные:**

name – Название насекомого (String)

speed – Скорость передвижения насекомого (Double)

color – Окраска насекомого (String)

eat – Используемая пища насекомого (String)

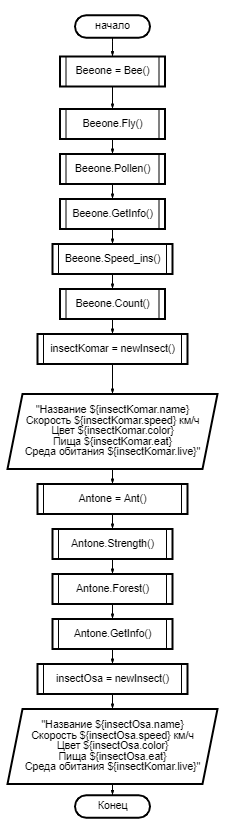
live – Среда обитания насекомого (String)

kind – Разновидность насекомого (String)

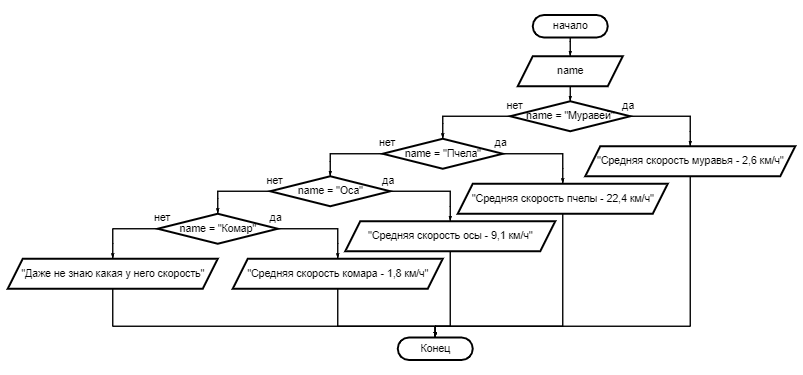
weight – Вес насекомого (Double)

**Блок-схемы**

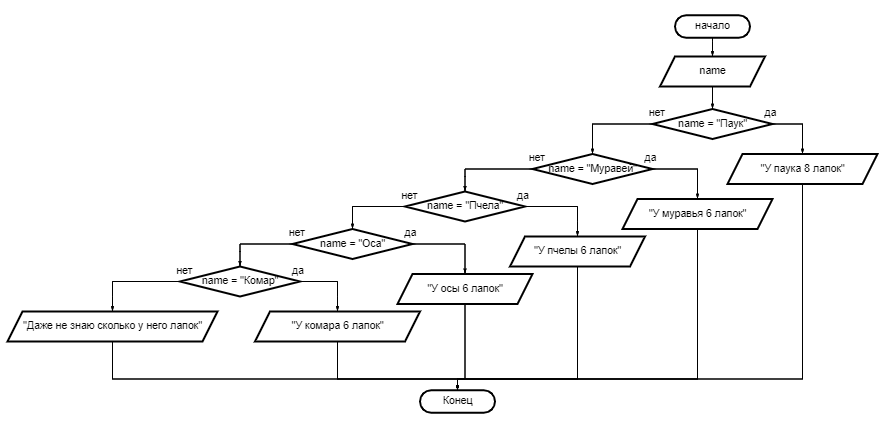
1)fun main()

****

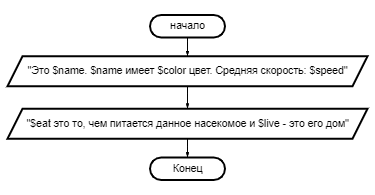
fun Speed\_ins()

****

fun Count()



fun GetInfo()(class Insects)

****

fun Strength()

****

fun Forest()

****

fun GetInfo()(class Ant)

****

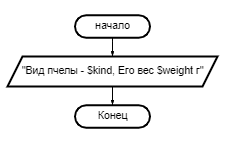
fun Fly()

****

fun Pollen()

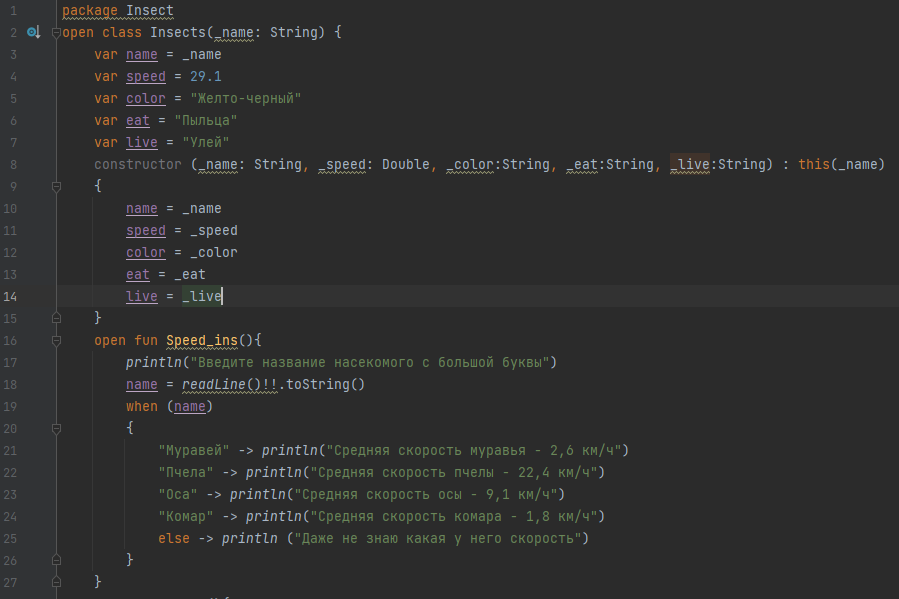
****

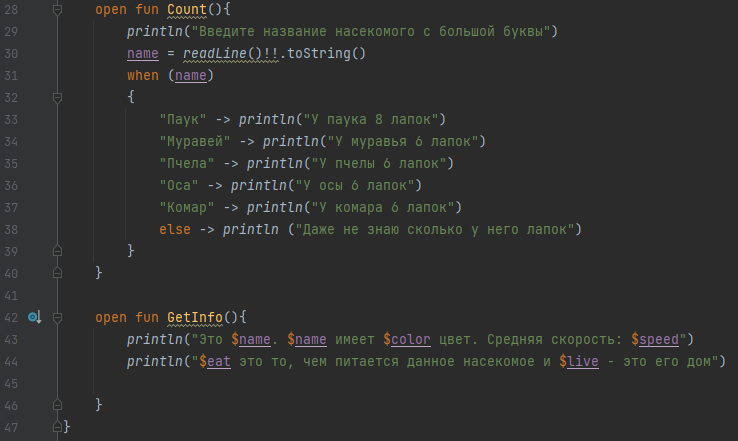
fun GetInfo()(class Bee)

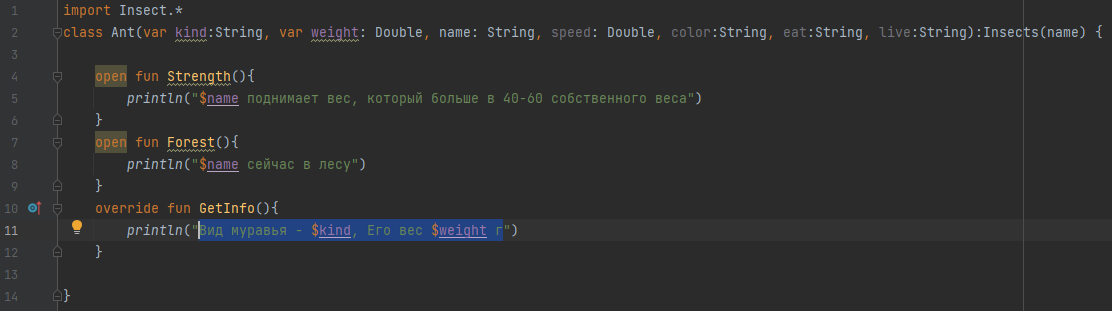
****

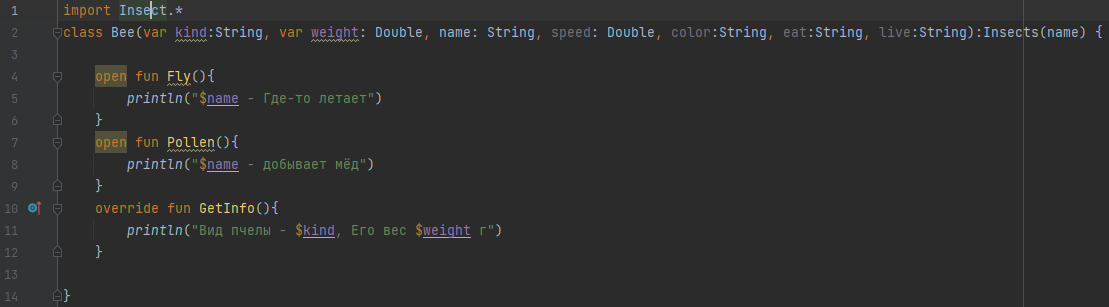
**Листинг программы**











**Тестовые ситуации**

1. Проверка на ввод названия насекомого (when, else)

**Задание 14:** Класс «Автомобиль»: марка, мощность, стоимость.

**Входные данные:**

brand – Марка автомобиля (String)

count – Количество автомобилей марки BMW (Int)

newprice – Цена за автомобиль марки BMW (Double)

**Выходные данные:**

brand – Марка автомобиля (String)

power – Мощность автомобиля (Double)

price – Стоимость автомобиля (Double)

model – Модель автомобиля (String)

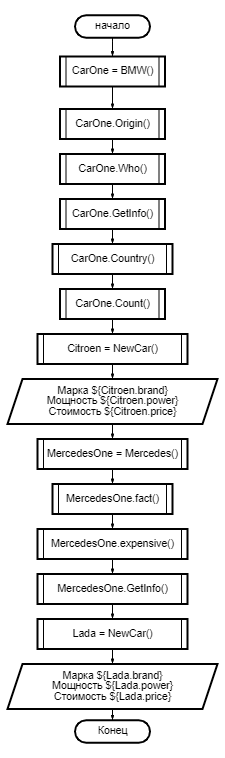
color – Цвет автомобиля (String)

count – Количество автомобилей марки BMW (Int)

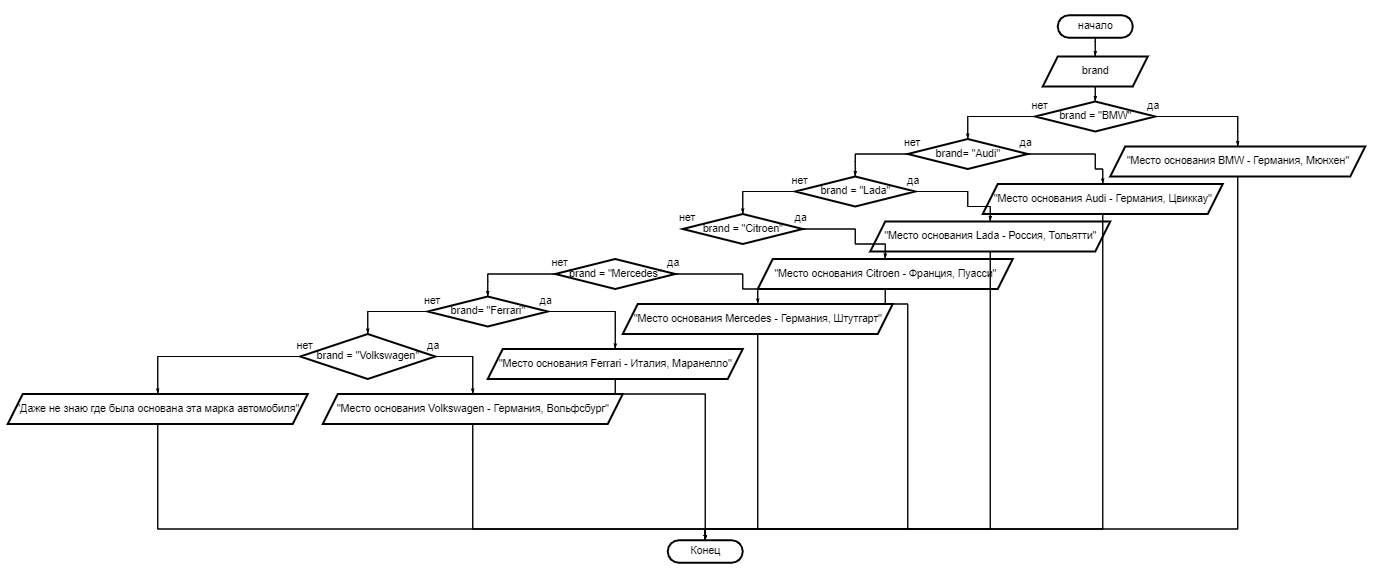
newprice – Цена за автомобиль марки BMW (Double)

**Блок-схемы**

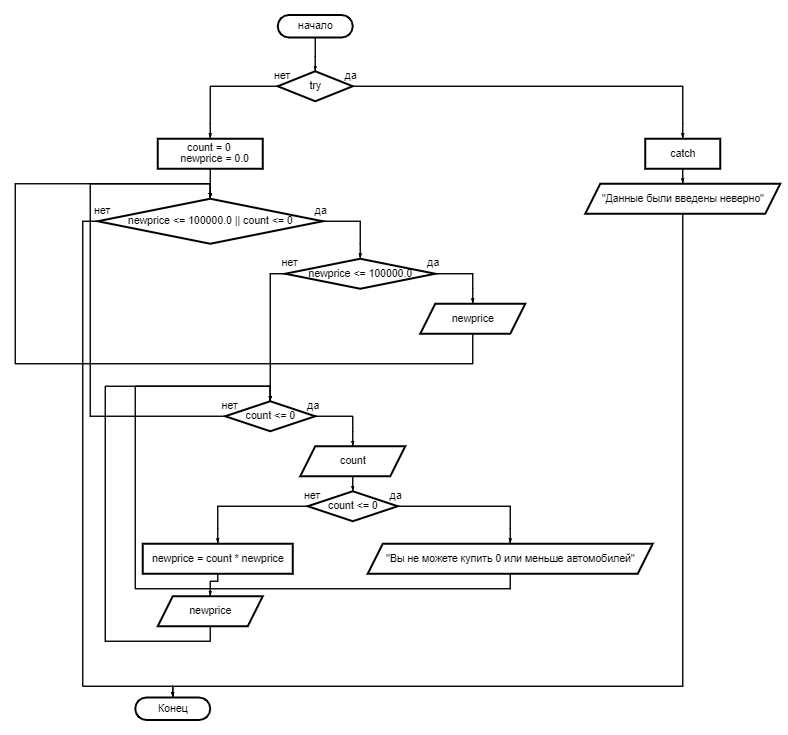
fun main()



fun Country()



fun Count()



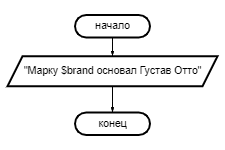
fun GetInfo()(class Cars)



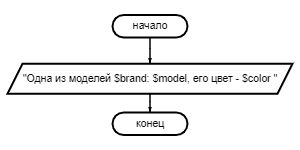
fun Origin()



fun Who()



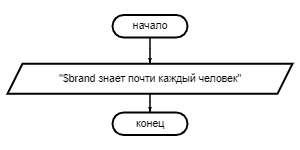
fun GetInfo()(class BMW)



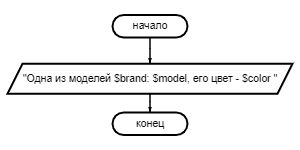
fun Fact()



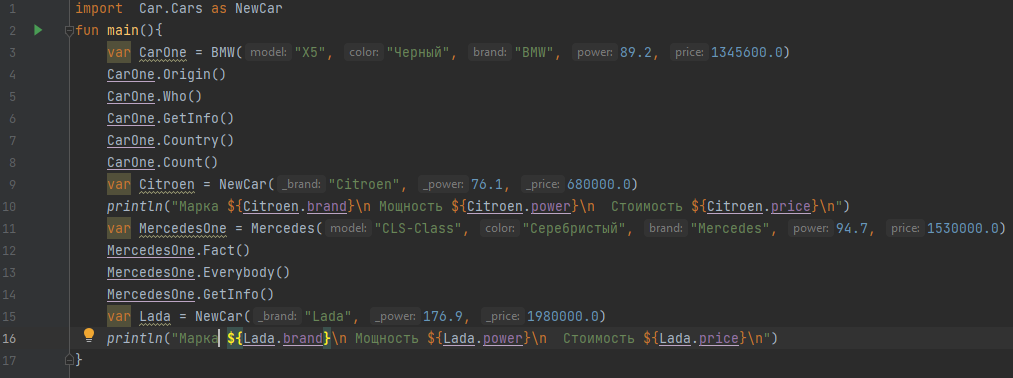
fun Everybody()

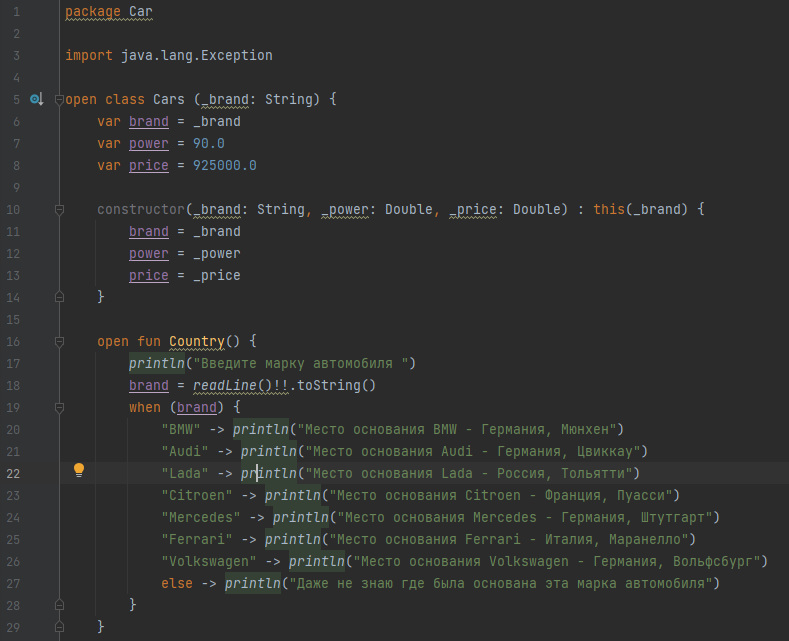


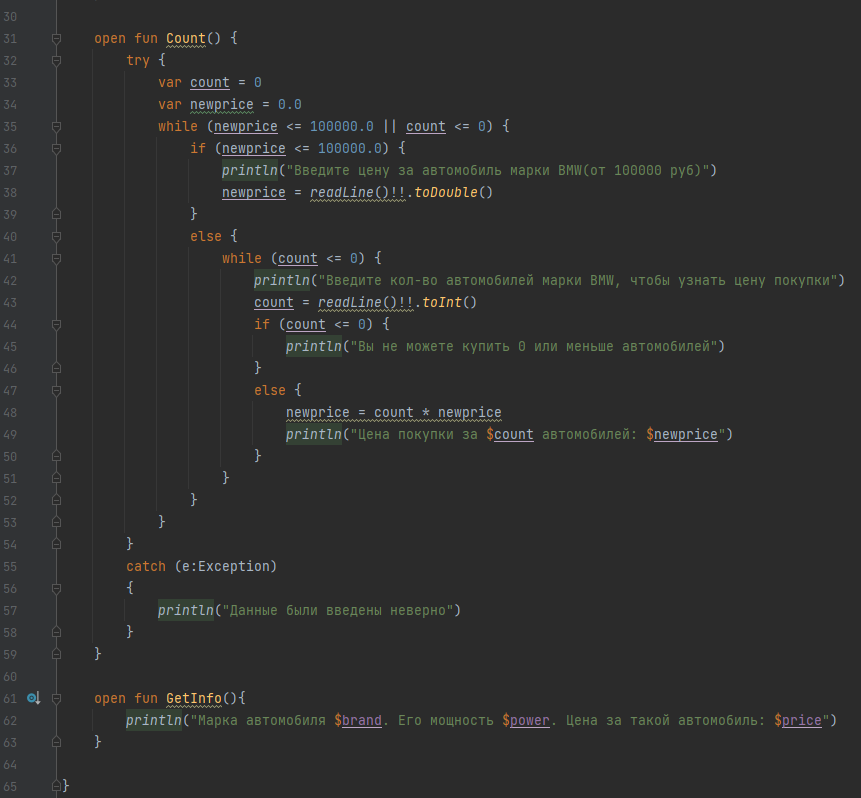
fun GetInfo()(class Mercedes)

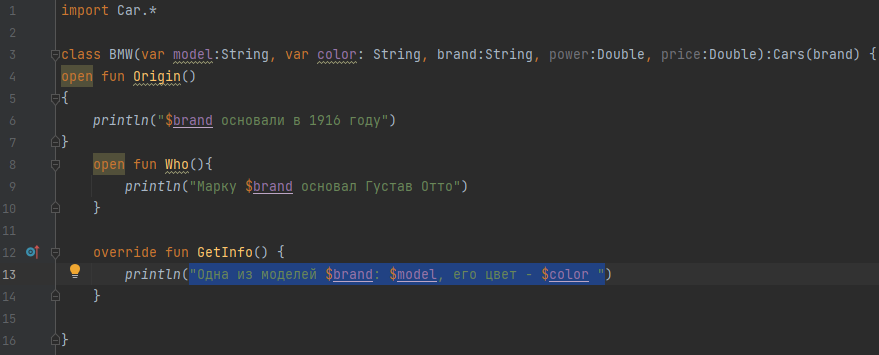


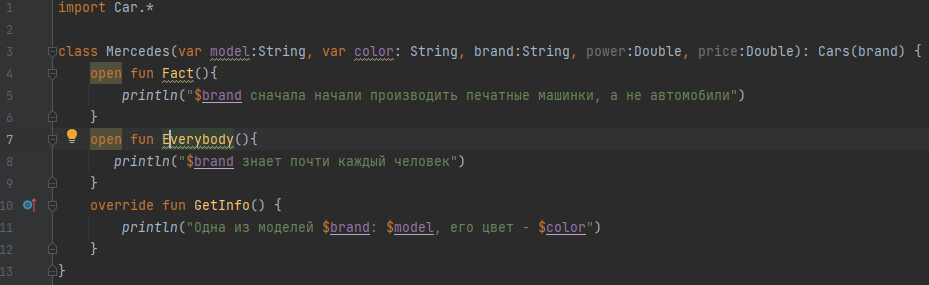
**Листинг программы**











**Тестовые ситуации**

1. Проверка на ввод названия марки (when, else)
2. Проверки на ввод чисел в определенном диапазоне (while)
3. Проверки на ввод чисел (try, catch)

**Вывод:** Я научился использовать классы-наследники с дополнительными конструкторами и функциями в Kotlin