Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования

Кафедра инженерной психологии и эргономики

Дисциплина: Современные языки программирования

Отчёт

По лабораторной работе №1

**ИСПОЛЬЗОВАНИЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ KOTLIN**

Вариант 7

Выполнила:

студентка группы 310101

Бычек Е. Н.

Проверил:

Усенко Ф. В.

Минск 2024

***Цель:*** Выполнить разработку приложения с использованием языка программирования Kotlin.

Добавить класс Трактор, наследующий от класса Автомобиль, обладающий методом убирать снег(). Добавить класс Водитель, управляющий классом Автомобиль.

Разработано приложение на языке программирования Kotlin. Создано 8 классов: Car, Tractor, Driver, Detail, Corpus, Engine, Wheel и Fuel. Программа отображает подробный процесс заведения трактора и уборки им снега. Осуществлена проверка на ввод некорректных данных через конструкцию if-else и нечисловых данных через try/catch. Результат выполнения работы программы представлен на рисунке 1.

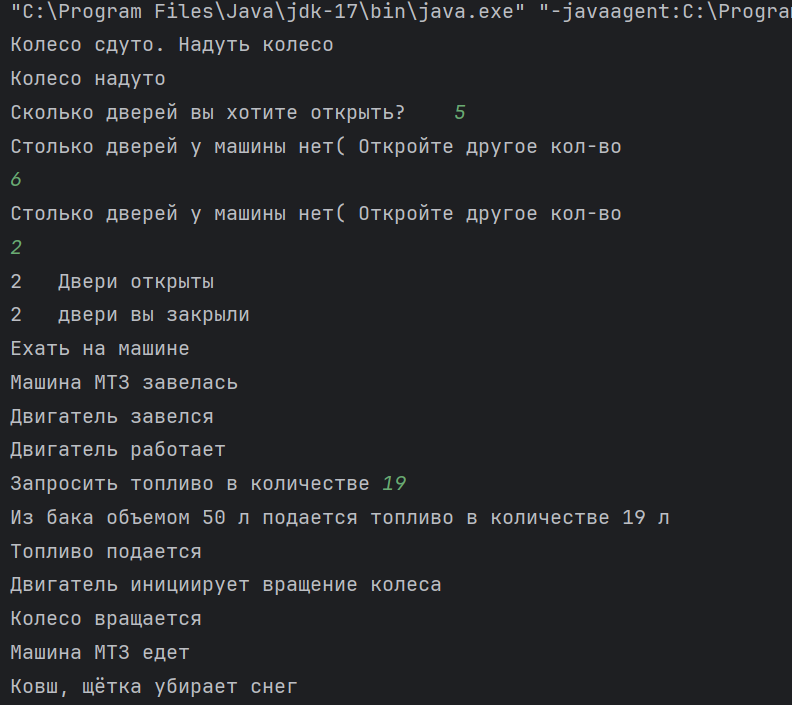


Рисунок 1 – Результат выполнения работы программы

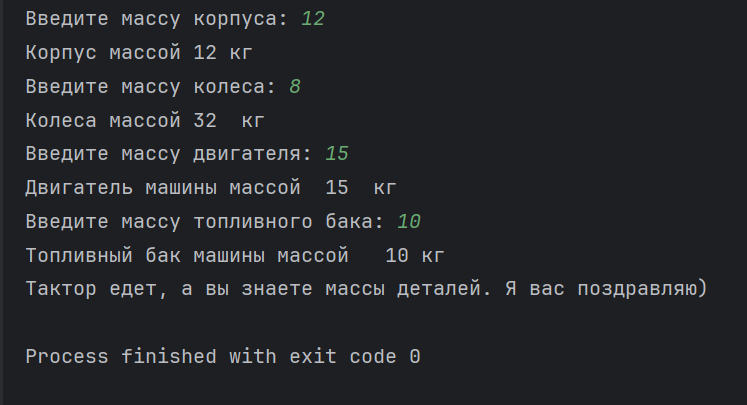


Рисунок 1 – лист 2

Так же создан Unit-test для проверки метода goByCar() класса Car. Результат выполнения работы Unit-test представлен на рисунке 2.

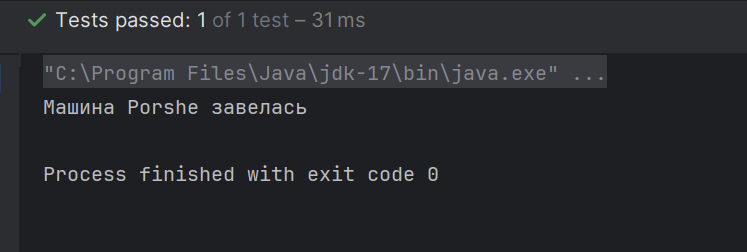


Рисунок 2 – Результат выполнения работы Unit-test для метода goByCar()

**Вывод:** В ходе лабораторной работы создано приложение на языке программирования Kotlin.

**Ответы на вопросы**

3. Классы в Kotlin объявляются с помощью использования ключевого слова *class*.

*class Person { /\*...\*/ }*

Объявление класса состоит из имени класса, заголовка (указания типов его параметров, основного конструктора и т.п) и тела класса, заключённого в фигурные скобки. И заголовок, и тело класса являются необязательными составляющими. Если у класса нет тела, фигурные скобки могут быть опущены.

6. Класс может быть объявлен как abstract со всеми или некоторыми его членами. Абстрактный член не имеет реализации в своём классе. Обратите внимание, что нам не надо аннотировать абстрактный класс или функцию словом open – это и так подразумевается.

10. Полная синтаксическая форма лямбда-выражений может быть представлена следующим образом:

*val sum: (Int, Int) -> Int = { x: Int, y: Int -> x + y }*

лямбда-выражение всегда заключено в скобки *{...}*;

объявление параметров при таком синтаксисе происходит внутри этих скобок и может включать в себя аннотации типов;

тело функции начинается после знака *->*;

если тип возвращаемого значения не *Unit*, то в качестве возвращаемого типа принимается последнее (а возможно и единственное) выражение внутри тела лямбды.

Если вы вынесите все необязательные объявления, то, что останется, будет выглядеть следующим образом:

*val sum = { x: Int, y: Int -> x + y }*