Министерство цифрового развития

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

Факультет Кибербезопасности и управления

Кафедра Управление в технических системах

**Отчёт**

**По лабораторной работе №3**

по дисциплине Прикладное программирование

**ВЫПОЛНИЛ**

студент ПрИ-22

Щелкунов Александр

**ПРОВЕРИЛ**

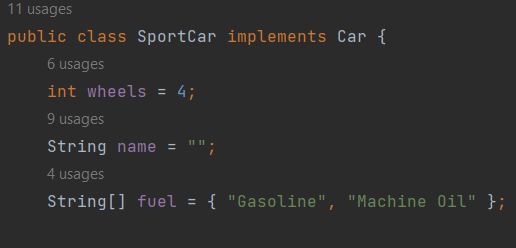
**Гильмуллин Рауль Ильнурович**

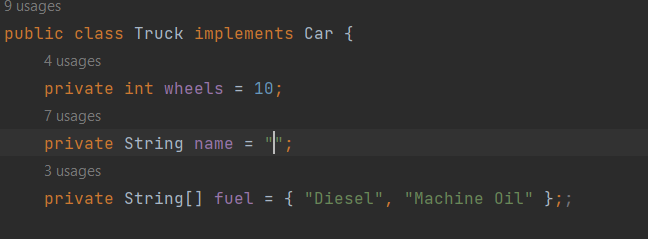
**Кадирова Валерия Анатольевна**

Самара

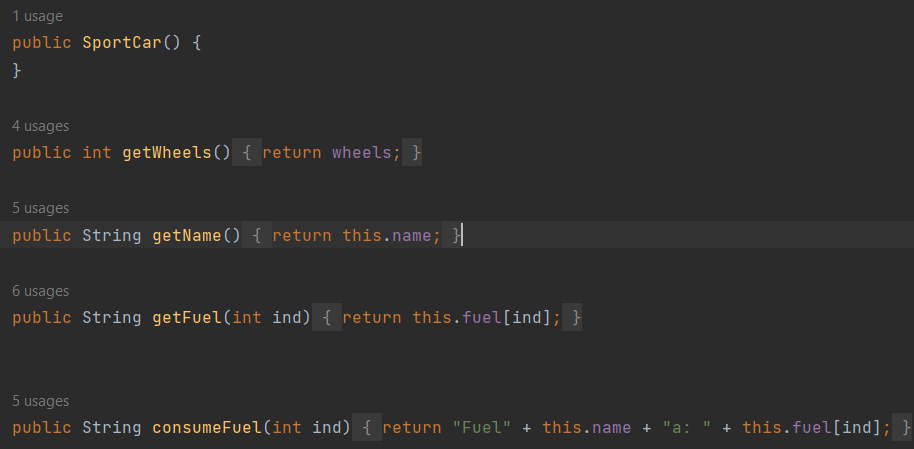
2024

-поле – массив, поле строкового типа, поле целого типа, возможно добавление других необходимых полей:

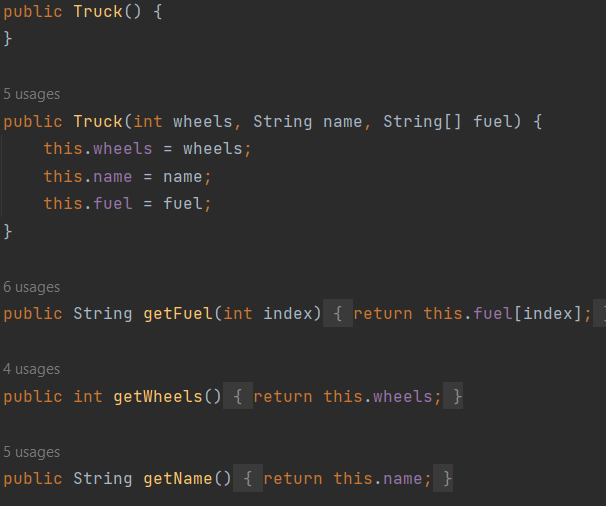


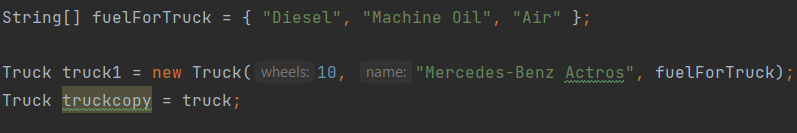


-конструктор по умолчанию, и конструктор с параметрами, позволяющими полностью инициализировать объект:

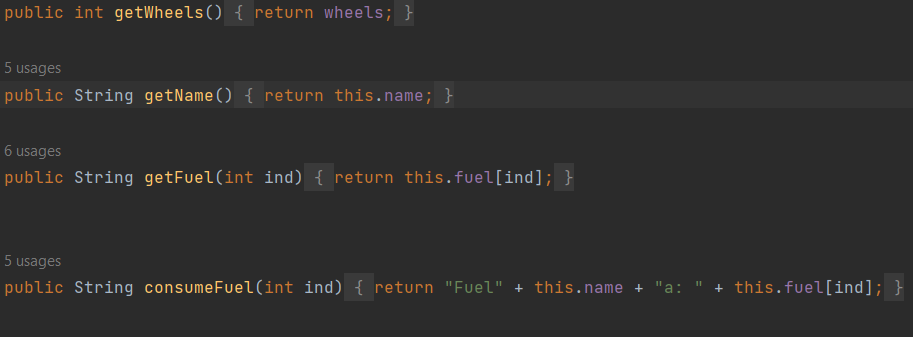


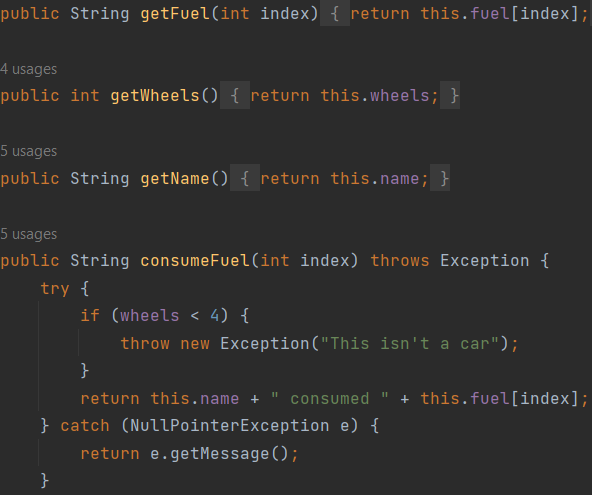




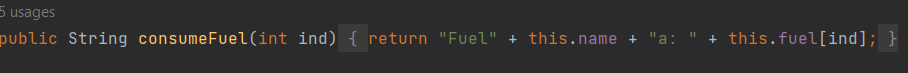


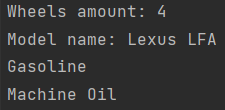
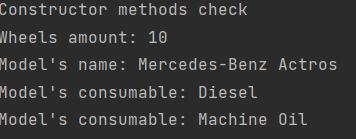
-методы доступа к элементам массивов и к полям:



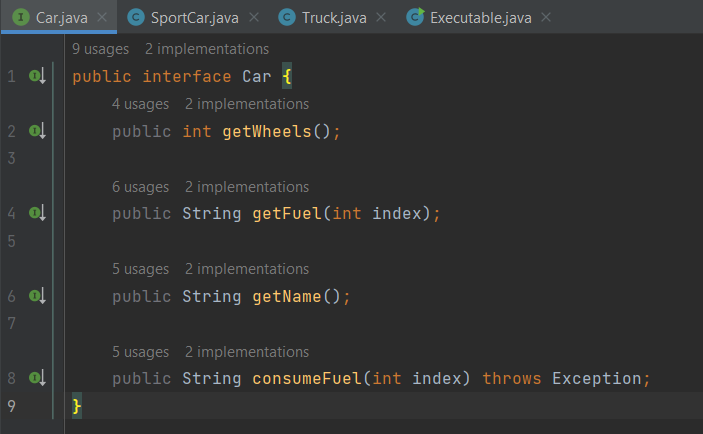


-функциональный метод, реализующий некоторую функцию от элементов массива и полей объектов классов (арифметическую, логическую, конкатенацию и т.п.):

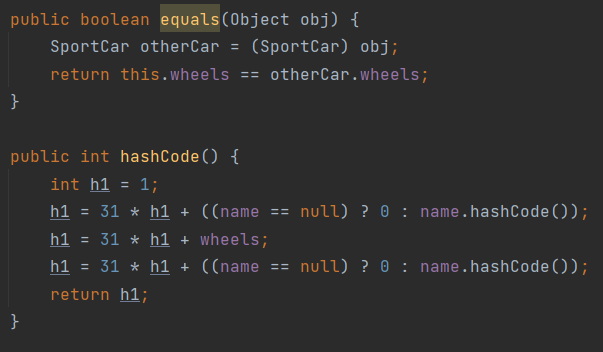


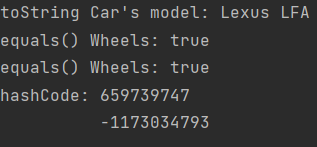
 

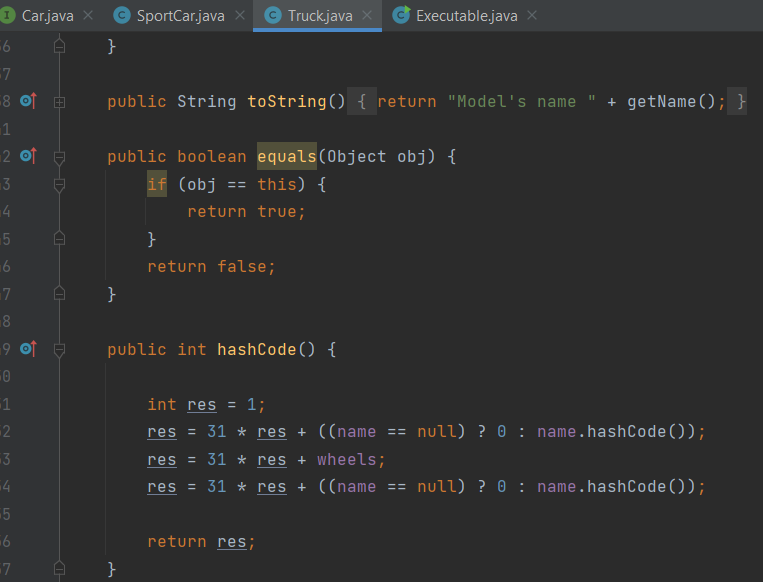
Описать интерфейс, задающий список сигнатур методов доступа к полям объектов и функционального метода, оба класса должны реализовывать интерфейс:

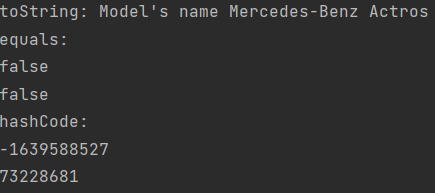


В классах корректно переопределить методы класса Object – toString(), equals(), hashCode():









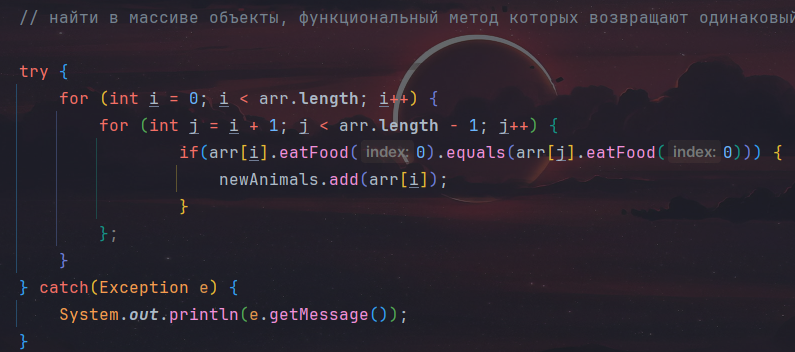
4. Организовать базу (массив) объектов типа интерфейс, которую заполнить объектами описанных типов вперемежку (по желанию пользователя). Выполнить следующие действия:

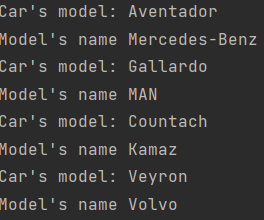
- вывести полную информацию обо всех объектах массива:

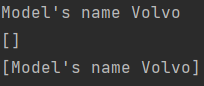
- найти в массиве объекты, функциональный метод которых возвращают одинаковый результат, поместить такие объекты в другой массив;:

- разбить исходный массив на два массива, в которых будут храниться однотипные элементы:

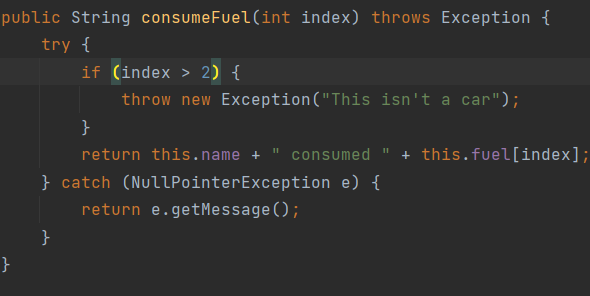


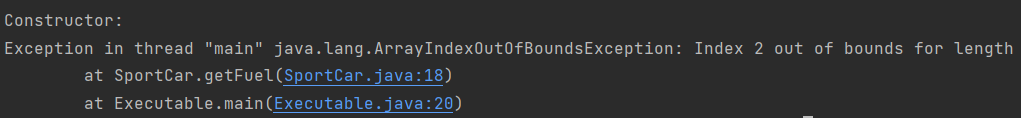






При описании классов описать собственные исключения - одно объявляемое (наследное от Exception) и одно необъявляемое (наследное от RuntimeException) исключения





Вопросы:

1) Состав класса в Java включает в себя поля (переменные), методы (функции), конструкторы, блоки инициализации, вложенные классы и интерфейсы.  
2) Модификаторы доступа определяют уровень доступности класса, поля или метода для других классов. В Java существуют четыре модификатора доступа: public, private, protected, default.  
3) Модификаторы полей в Java определяют уровень доступности к полю. Они могут быть такими же, как у модификаторов доступа: public, private, protected, default, а также static, final и transient.  
4) Модификаторы методов в Java определяют поведение метода. Они могут быть такими же, как у модификаторов доступа: public, private, protected, default, а также static, final, abstract, synchronized и другие.  
5) Одномерные массивы в Java это структуры данных, содержащие элементы одного типа данных. Для обработки одномерных массивов в Java используются циклы, как например for или foreach.  
6) Многомерные массивы в Java это массивы, содержащие другие массивы. Для обработки многомерных массивов в Java используются вложенные циклы.  
7) Класс Object является корневым классом в Java, от которого наследуются все остальные классы. Он имеет методы, которые переопределяются в других классах, такие как toString(), equals() и hashCode().  
8) Исключения в Java это ошибки, возникающие во время выполнения программы. Для отлова исключений используется конструкция try-catch, которая позволяет перехватывать и обрабатывать исключения.  
9) Объявляемые исключения в Java это исключения, которые должны быть указаны в сигнатуре метода. Для работы с ними используется конструкция throws.  
10) Необъявляемые исключения в Java это ошибки, которые могут возникнуть в процессе выполнения программы, и не требуют объявления в сигнатуре метода.  
11) Наследование в Java позволяет создавать иерархию классов, где подкласс наследует поля и методы суперкласса.  
12) Наследование конструкторов в Java позволяет подклассу вызывать конструкторы суперкласса, используя ключевое слово super.  
13) Наследование полей и методов в Java позволяет подклассу использовать поля и методы суперкласса.  
14) Ключевое слово super() в Java используется для вызова конструктора суперкласса из конструктора подкласса.  
15) Операции проверки и преобразования типов в Java позволяют проверить тип объекта или преобразовать его к другому типу.  
16) Завершенные классы и методы - это классы и методы, которые нельзя переопределить в других классах.  
17) Абстрактные методы и классы в Java позволяют определить методы без их реализации.  
18) Интерфейс в Java это контракт, определяющий методы, которые должны быть реализованы классом. Он содержит только константы и абстрактные методы.  
19) Реализация интерфейса в Java позволяет классу реализовать методы, определенные в интерфейсе.  
20) Наследование для интерфейсов в Java позволяет интерфейсу наследовать другой интерфейс.