МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА №53

ОТЧЕТ

ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ассистент |  |  |  | А. К. Малышев |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЁТ ПО КУРСОВОЙ РАБОТЕ  «Разработка архитектуры классов для просмотра характеристики автомобилей»  по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ |
|  |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 5836 |  |  |  | Д. А. Храмченко |  |
|  |  |  | подпись, дата |  | инициалы, фамилия |  |

Санкт-Петербург 2018 г.

Оглавление

## Анализ поставленной задачи: функциональность разрабатываемой библиотеки и назначение ее использования

## Диаграмма классов

## Описание программного кода классов

## Тестирование функциональности классов

## Пример использования библиотеки для построения интерфейса прикладного решения

## Руководство пользователя

## Заключение

## Библиографический список

## Приложения

## Анализ поставленной задачи: функциональность разрабатываемой библиотеки и назначение ее использования.

Требуется разработать библиотеку классов для просмотра характеристик грузового и легкового транспорта. Для этого необходимо определить иерархию классов, которой может воспользоваться пользователь: транспорт, легковой автомобиль, грузовой автомобиль. Модель должна позволять пользователю создавать, просматривать и удалять легковые и грузовые автомобили.

Объекты класса «легковой автомобиль» должны иметь следующие поля:

* Бренд
* Страна происхождения бренда
* Дата создания бренда
* Цвет
* Цена
* Масса
* Год выпуска
* Картинка
* Тип кузова

Объекты класса «легковой автомобиль» должны иметь следующие методы:

* Строковое представление объекта
* Корректность объекта
* Конструктор без параметров
* Конструктор с параметрами

Объекты класса «грузовой автомобиль» должны иметь такие поля:

* Бренд
* Страна происхождения бренда
* Дата создания бренда
* Цвет
* Цена
* Масса
* Год выпуска
* Картинка
* Длина прицепа

Объекты класса «грузовой автомобиль» иметь методы:

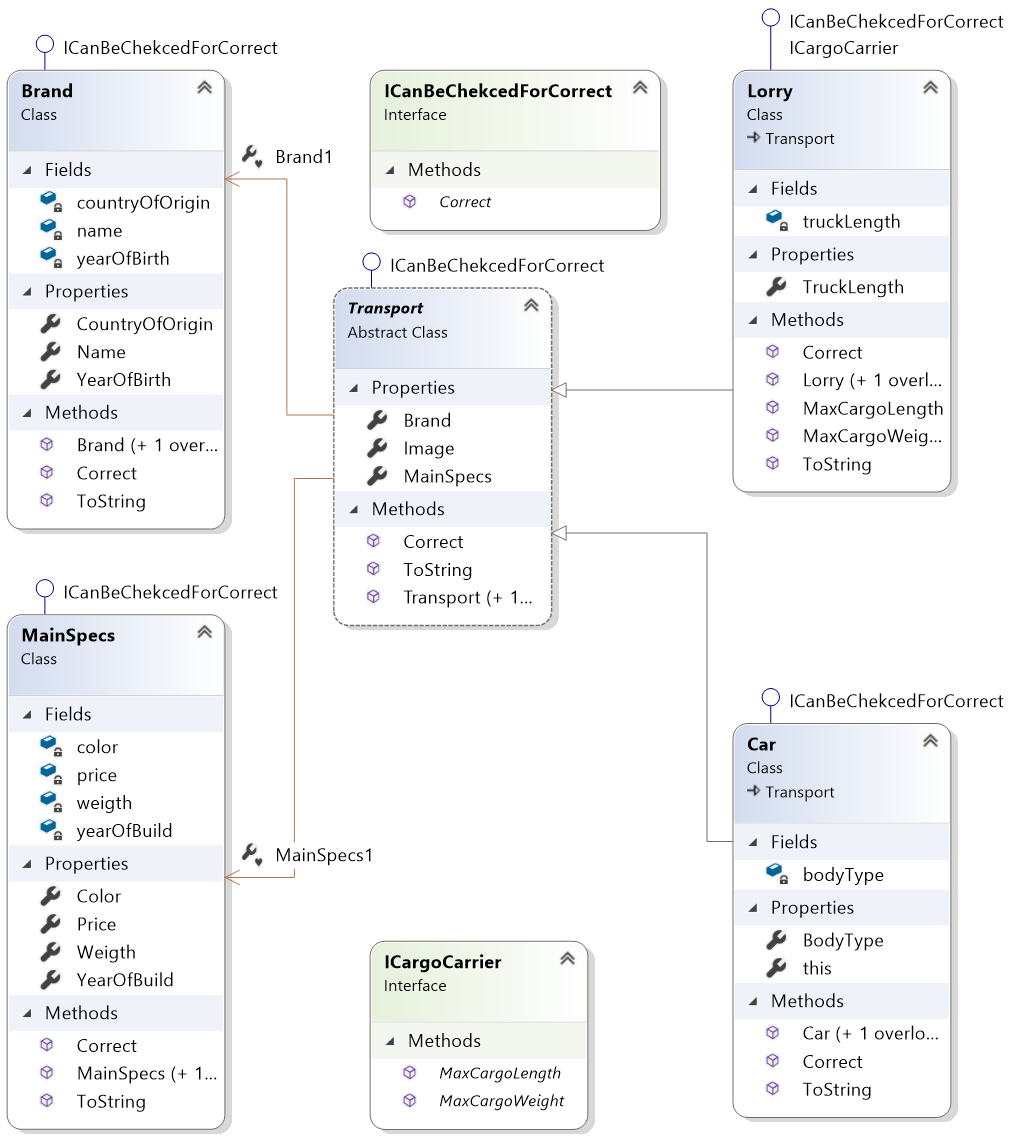
* Строковое представление объекта
* Корректность объекта
* Конструктор без параметров
* Конструктор с параметрами
* Вычисление максимальной массы грузы
* Вычисление максимальной длины груза

Для работы с этими объектами необходимо создать абстрактный базовый класс «Транспорт», в котором будет находиться информация о бренде, основные характеристики автомобиля, картинка, методы строкового представления и проверки корректности объектов. Также нужно создать и классы для хранения информации о бренде и основных характеристик автомобиля.

В классах описать следующие элементы:

* Поля;
* Индексатор для доступа к свойству по имени свойства(для легковых);
* Конструкторы с параметрами и конструкторы по умолчанию;
* Свойства для установки и получения всех характеристик, информации об автомобиле и картинки.
* Методы строкового представления объектов, методы проверки корректности
* Методы для вычисления максимальной массы и максимальной длины груза(для грузового)

## Диаграмма классов.

****

## Описание программного кода классов.

Интерфейс ICanBeCheckedForCorrect создан для проверки объекта на корректность и содержит метод correct, который возвращает true или false.

Интерфейс ICargoCarrier создан для вычисления максимальной массы груза и максимальной длины грузы.

Методы

* MaxCargoLength – возвращает максимальную длину груза
* MaxCargoWeight – возвращает максимальную массу груза

Класс Brand хранит информацию о бренде автомобиля и реализует интерфейс ICanBeCheckedForCorrect.  
Поля

* Имя
* Страна происхождения
* Год создания

Так же он содержит свойства для чтения и записи в/эти поля.

Методы

* Correct для проверки на корректность
* ToString возвращает строковое представление объекта

Конструкторы

* Конструктор без параметров
* Конструктор с параметрами, позволяющий инициализировать все поля объекта через свойства

Класс MainSpecs хранит основные характеристики автомобиля.

Поля

* Цвет
* Цена
* Масса
* Год выпуска

Так же он содержит свойства для чтения и записи в/эти поля.

Методы

* Correct для проверки на корректность
* ToString возвращает строковое представление объекта

Конструкторы

* Конструктор без параметров
* Конструктор с параметрами, позволяющий инициализировать все поля объекта через свойства

Абстрактный класс Transport реализует интерфейс ICanBeCheckedForCorrect.

Transport связан отношением ассоциации с классами Brand и MainSpecs.

Свойства

* Бренд типа Brand хранит всю информацию о бренде
* MainSpecs типа MainSpecs хранит основные характеристики автомобиля
* Image типа Image хранит изображение автомобиля

Методы

* Correct для проверки на корректность
* ToString возвращает строковое представление объекта

Конструкторы

* Конструктор без параметров
* Конструктор с параметрами, позволяющий инициализировать все поля объекта через свойства

Класс Car создан для легковых автомобилей. Наследник класса Transport.  
Pеализует интерфейс ICanBeCheckedForCorrect.

Поля

* Тип кузова

Свойства

* BodyType для записи/ чтения из/в поле для типа кузова
* Индексатор доступа к свойствам объекта через их имя

Методы

* Correct для проверки на корректность
* ToString возвращает строковое представление объекта

Конструкторы

* Конструктор без параметров
* Конструктор с параметрами, позволяющий инициализировать все поля объекта через свойства

Класс Lorry создан для грузовых автомобилей. Наследник класса Transport.  
Pеализует интерфейс ICanBeCheckedForCorrect и ICargoCarrier.

Поля

* Длина прицепа

Свойства

* Для записи и чтения в/из поле для длины прицепа

Методы

* Correct для проверки на корректность
* ToString возвращает строковое представление объекта
* MaxCargoLength – возвращает максимальную длину груза
* MaxCargoWeight – возвращает максимальную массу груза

Конструкторы

* Конструктор без параметров
* Конструктор с параметрами, позволяющий инициализировать все поля объекта через свойства

## Тестирование функциональности классов.

1. Пример использования библиотеки для построения интерфейса прикладного решения.

## Руководство пользователя.

## Заключение

1. Библиографический список

## Приложения

**Приложение 1. Листинг программы**