Міністерство освіти і науки України

Національний технічний університет України „КПІ”

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки

інформації та управління

**ЗВІТ**

з основ Web-програмування

Лабораторна №2

на тему :

„ Приведення типів. Перевантаження операторів ”

|  |  |  |  |
| --- | --- | --- | --- |
| **Виконав студент** |  | Гасс Леонід Едуардович  Група ІП-61  Курс 2  Кафедра АСОІУ  Номер залікової книжки IP6105 |  |
|  |  |  |  |
|  |  |  |  |

Київ 2018

ЗМІСТ

[1 Постановка задачі 3](#_Toc515279669)

[2 Результати роботи програми 4](#_Toc515279670)

# Постановка задачі

**Варіант 3**

Створити абстрактний клас Vehicle. На його основі реалізувати класи Car (автомобіль), Bicycle (велосипед) і Lorry (вантажівка).

Класи повинні мати можливість задавати і отримувати параметри засобів пересування (ціна, максимальна швидкість, рік випуску і т.д.). Поряд із загальними полями і методами, кожен клас повинен містити і специфічні для нього поля.

Створити клас Garage, що містить масив / параметризрвані колекцію об'єктів цих класів в динамічній пам'яті.

Передбачити можливість виведення характеристик об'єктів списку.

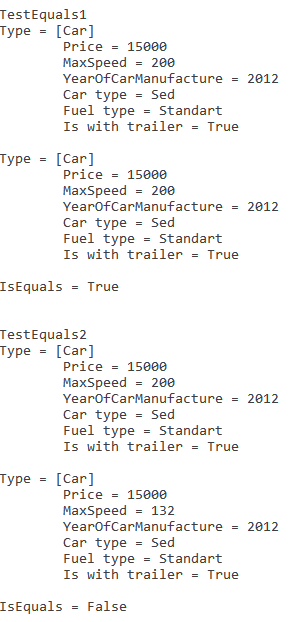
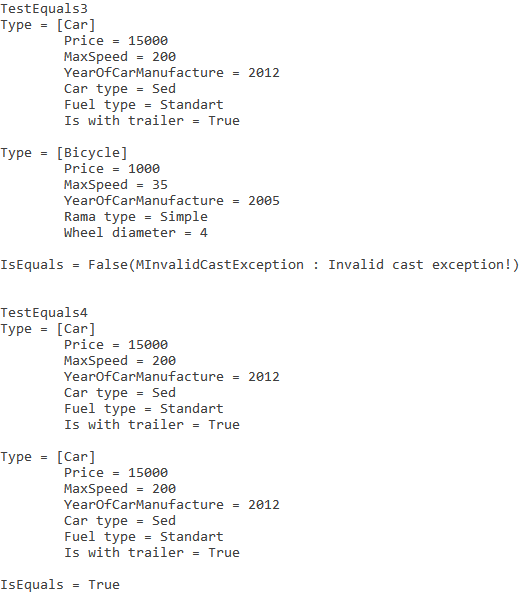
# Результати роботи програми

Під час роботи було створено абстрактний клас Vehicle, що описує абстракний транспортний засіб за допомогою загальних атрибутів, таких як price, maxSpeed, yearOfCarManufacture та містить абстрактні методи DeepCopy(), GetProperties() для зручності роботи з сімейством об’єктів та загальні методи, що працюють в усіх нащадках однаково(Напр. метод ToString() визначений лише в Vehicle та повертає результат абстрактного методу GetProperties(), що має власну реалізаціє в кожному класі-нащадку, таким чином ToString() працює для всіх в залежності від методу GetProperties()). В усіх інших класах-нащадках Vehicle, були написані свої атрибути, що описують певний клас та були перевизначені методи такі як Equals(), GetHashCode() та оператори рівності та нерівності для зручності роботи з колекціями цих обьектів.

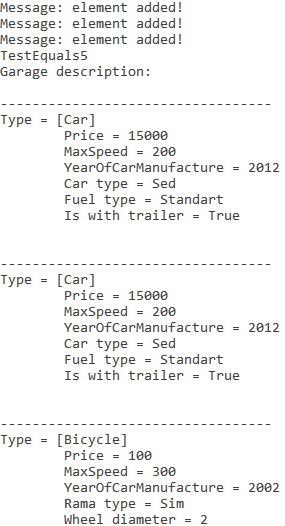
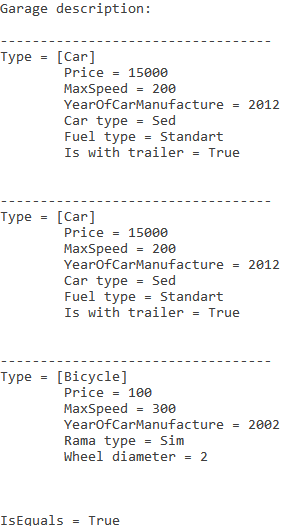
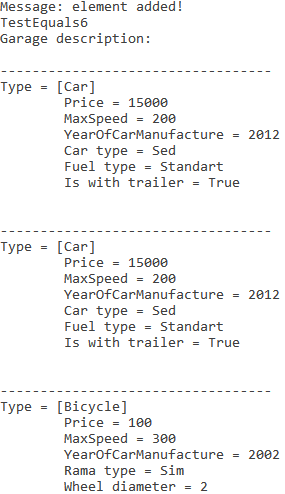
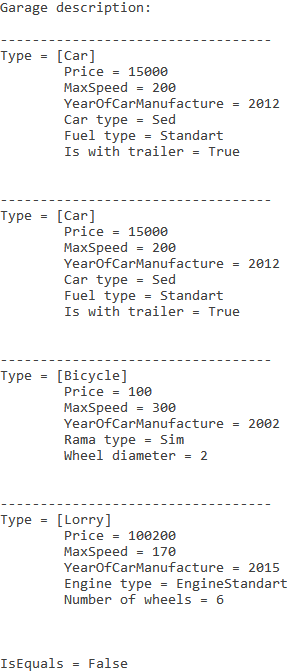
Для власної реалізації ArrayList було визначено клас-обгортку WrapperList, що містить в собі звичайний список с тими ж методами, та що містить івентами з делегатами для відстежування виконання цих методів – Add(), Remove() та Update(). Було написано клас Garage() що містить в собі об’єкт типу WrapperList та реалізує методи для роботи з ним та керування автомобілями в ньому. За допомогою гарної реалізації класс WrapperList, ми отримали невеликий класс Garage, містить майже в кожному методі виклик методу цього классу(WrapperList).

В кожному методі, що виконує дії, що можуть привести до непердбачених подій (порівняння об’экту з іншим типом, вихід за межі массиву і так далі) було передбачено кидання виключень.

Приклади виконання програми

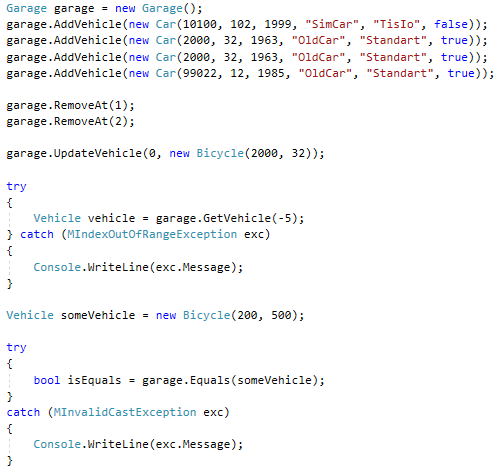
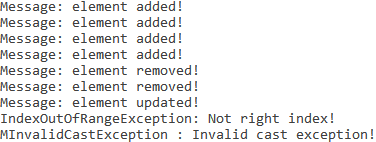
 

Порівняння об’єктів класів-нащадків Vehicle між собою

Порівняння об’єктів типу Garage між собою

(при додаванні(видаленні/оновленні) транспортних с гаражу приходить повідомлення)

Приклад реагування на додавання 4 т.з., видалення 2 та оновлення одного(events реагують на них) та спостерігання кидання виключення при спробі взяття елементу з колекції з індексом -5 та перевірці рівності гаражу та т.з.(транспортного засобу).