**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение высшего образования**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Инженерная школа информационных технологий и робототехники Отделение информационных технологий**

**Информатика и вычислительная техника**

Отчет по Лабораторной работе 3

Создание объектной модели предметной области

Выполнил

Студент группы 8В23 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Балашов А.Д.

Проверил

Доцент ОИТ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Хамухин А. А.

Томск 2024

**Цель работы:**

Получить опыт практической работы в создании иерархий классов предметной области с помощью UML диаграмм.

**Задание**

Необходимо описать иерархию классов предметной области в виде UML диаграммы и классов на языке C# в соответствии с выбранным вариантом задания (см. приложение).

Необходимо выполнить только первую часть задания, т.е. логику по расчету значений и вывод данных на экран выполнять не надо, только описание классов.

Требования и рекомендации:

1. Иерархия классов должна состоять минимум из 3 уровней;
2. Каждый класс должен содержать уникальный набор атрибутов;
3. Классы должны быть оформлены в виде DLL библиотеки в консольном проекте;
4. При именовании переменных и описании классов необходимо придерживаться соглашения о кодировании (см. п. 3.5).

**Процесс выполнения**

Составил ассоциативный ряд вида: транспорт (какой?) – наземный/воздушный/водный (тип?) – общественный/частный.  
Каждый элемента такой структуры имеет наследование по иерархии.

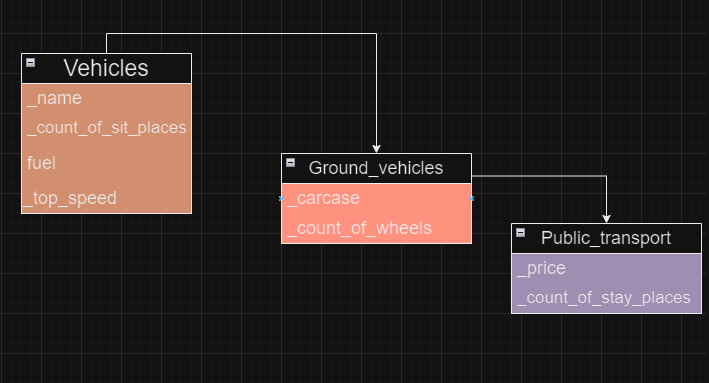
****

Рисунок 1 – UML диаграмма

После составление UML написал код на предлагаемом ЯП C#:

namespace Vehicles\_

{

class Vehicles

{

private string \_name;

private int \_count\_of\_sit\_places;

private string \_fuel;

private int \_top\_speed;

public string Name

{

get { return \_name; }

set { \_name = value; }

}

public int Count\_of\_sit\_places

{

get { return \_count\_of\_sit\_places; }

set { \_count\_of\_sit\_places = value; }

}

public string Fuel

{

get { return \_fuel; }

set { \_fuel = value; }

}

public int Top\_speed

{

get { return \_top\_speed; }

set { \_top\_speed = value; }

}

};

class Ground\_vehciles : Vehicles

{

private string \_carcase;

private int \_count\_of\_wheels;

public string Carcase

{

get { return \_carcase; }

set { \_carcase = value; }

}

public int Сount\_of\_wheels

{

get { return \_count\_of\_wheels; }

set { \_count\_of\_wheels = value; }

}

};

class Public\_Transport : Ground\_vehciles

{

private double \_price;

private int \_count\_of\_stay\_places;

public double Price

{

get { return \_price; }

set { \_price = value; }

}

public int Count\_of\_stay\_places

{

get { return \_count\_of\_stay\_places; }

set { \_count\_of\_stay\_places = value; }

}

};

}

**Ответы на контрольные вопросы**

**Что такое наследование?**

Наследование - это механизм в ООП, который позволяет классу наследовать свойства и методы другого класса. Класс, который наследует свойства и методы, называется подклассом или дочерним классом, а класс, чьи свойства и методы наследуются, называется суперклассом или родительским классом. Наследование позволяет повторно использовать код и создавать иерархии классов.

**Что такое инкапсуляция?**

Инкапсуляция - это один из принципов ООП, который позволяет скрывать внутреннюю реализацию объекта от внешнего мира и предоставлять доступ к ней только через определенные методы и свойства. Таким образом, данные и методы, работающие с этими данными, объединяются в единый объект, что обеспечивает безопасность и упрощает использование кода.

**Что такое полиморфизм?**

Полиморфизм - это возможность объектов с одинаковым интерфейсом иметь различную реализацию. Это означает, что один и тот же метод может работать по-разному в разных классах. Полиморфизм позволяет создавать гибкие и расширяемые системы, где различные объекты могут вести себя по-разному при вызове одного и того же метода.

**Что такое абстрактный класс?**Абстрактный класс - это класс, который содержит хотя бы один абстрактный метод. Абстрактный метод - это метод, который объявлен, но не реализован в абстрактном классе. Абстрактный класс нельзя инстанциировать напрямую; он служит в основном для создания общего интерфейса для всех его подклассов. Подклассы должны реализовать все абстрактные методы абстрактного класса или быть также абстрактными.

**Вывод**

В ходе выполнения лабораторной работы был получен опыт практической и теоретической работы в составлении UML, написании описывающего UML кода на ЯП C#.

**Список используемых источников:**

1. Джим Арлоу, Айла Нейштадт UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование, СимволПлюс, 2007
2. Дж. Рамбо, М. Блаха UML 2.0. Объектно-ориентированное моделирование и разработка, Питер, 2007
3. StarUML. Руководство пользователя [Электронный ресурс]. — Режим доступа: http://staruml.sourceforge.net/docs/user-guide(ru)/user-guide.pdf, свободный.
4. Блинов И.Н, Романчик В.С. Java 2. Практическое руководство. — Мн.: УниверсалПресс, 2005. — 400 с.
5. Диаграммы классов UML. Логическое моделирование [Электронный ресурс]. — Режим доступа: <http://www.informicus.ru/default.aspx?SECTION=6&id=73&subdivisionid=3>, свободный.