**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

Системы контроля версий

по дисциплине «Технологии программирования»

Выполнил

Студенты группы 8В12 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Вавилов М.А.

Проверил

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

Томск 2023

**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.

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.

**Вариант №3**

**Электрик.** Определить иерархию электроприборов. Включить некоторые в розетку. Посчитать потребляемую мощность

**Ход работы**

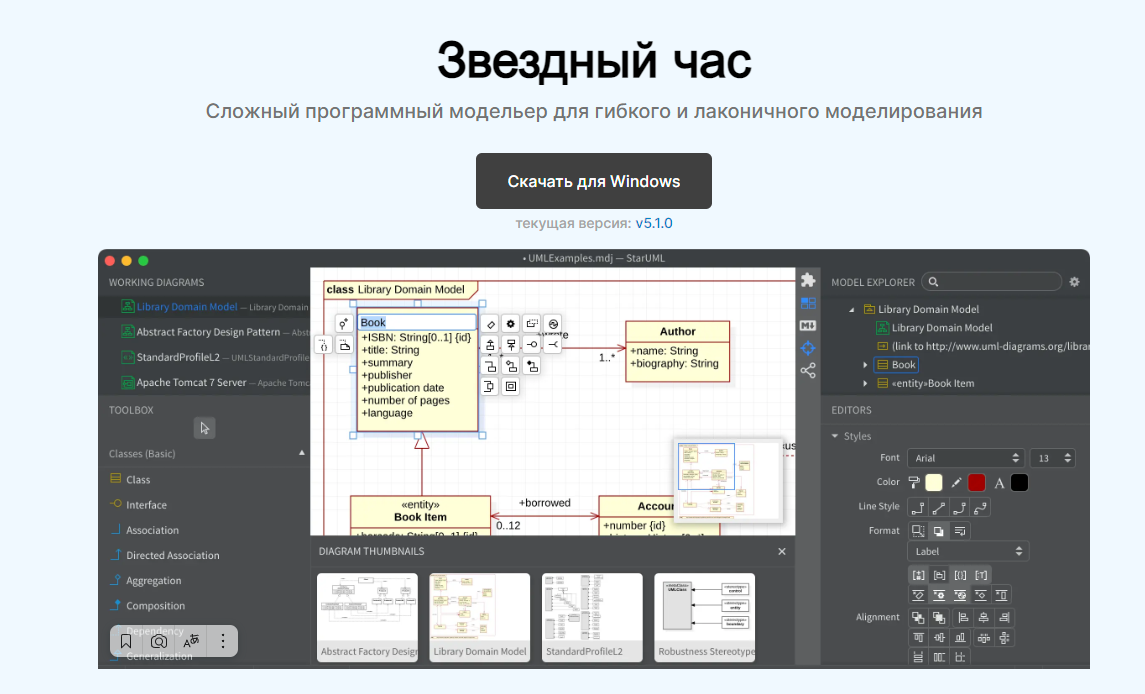
****

Рисунок 1 – Скачиваем StarUML.

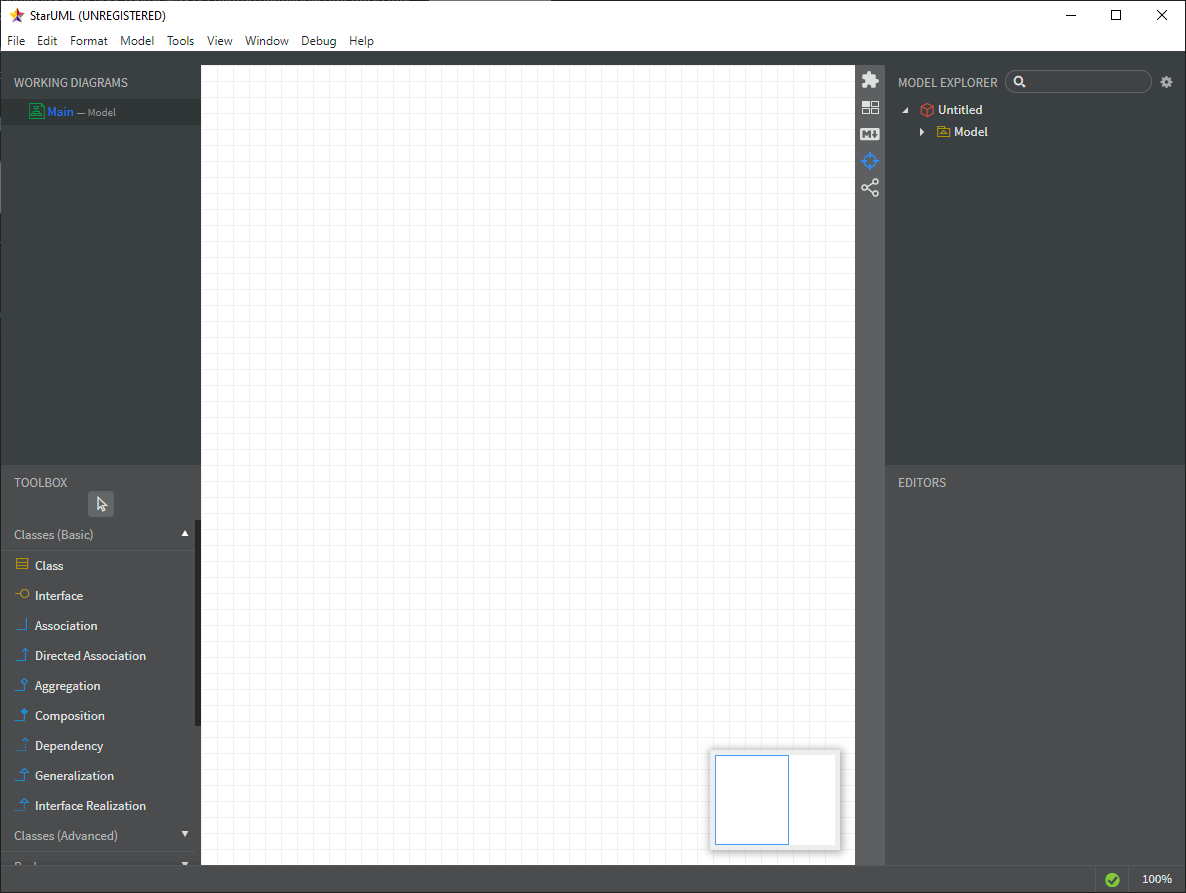


Рисунок 2 – Создаем новый проект в StarUML.

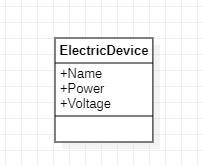


Рисунок 3 – Создаем класс ElectricDevice и задаем его атрибуты.

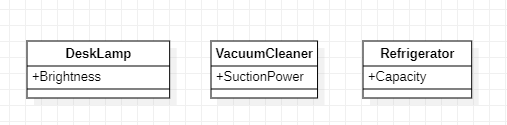


Рисунок 4 – Создаем подклассы.

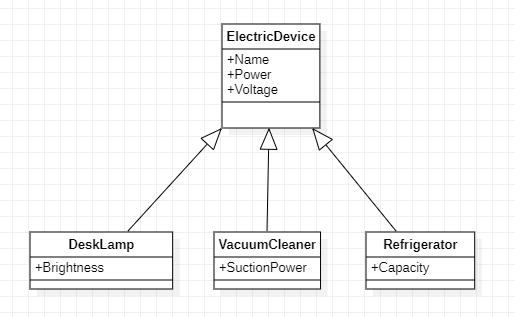
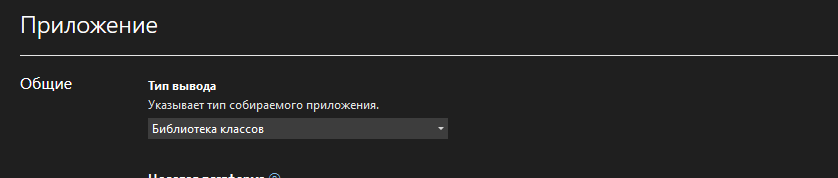
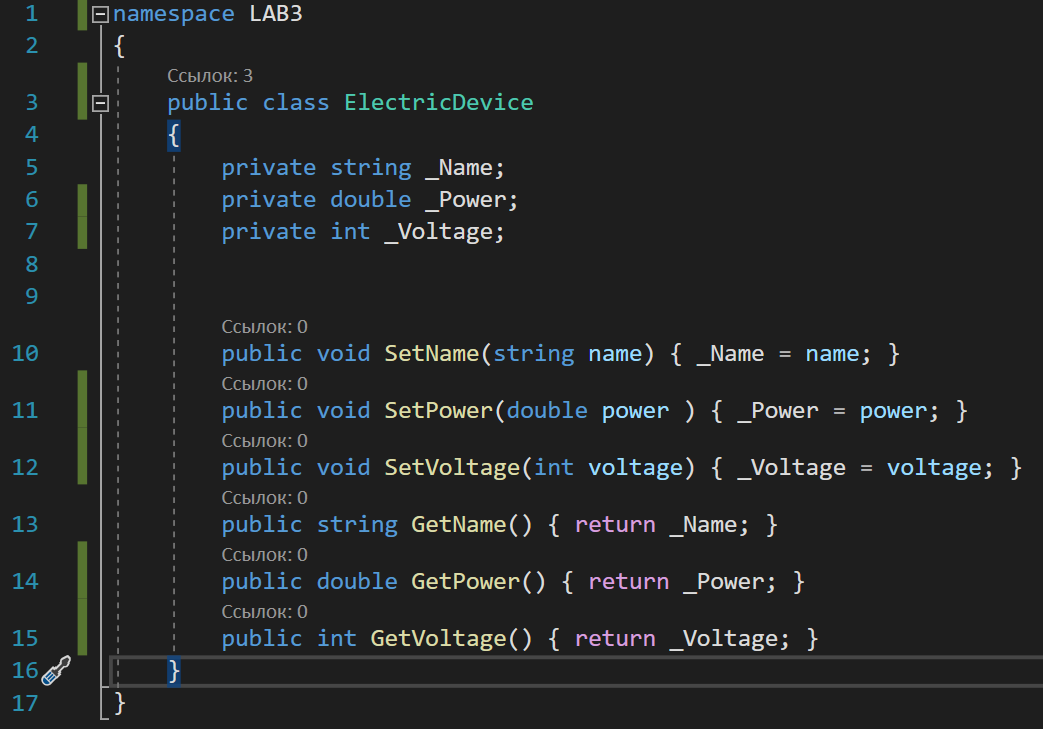
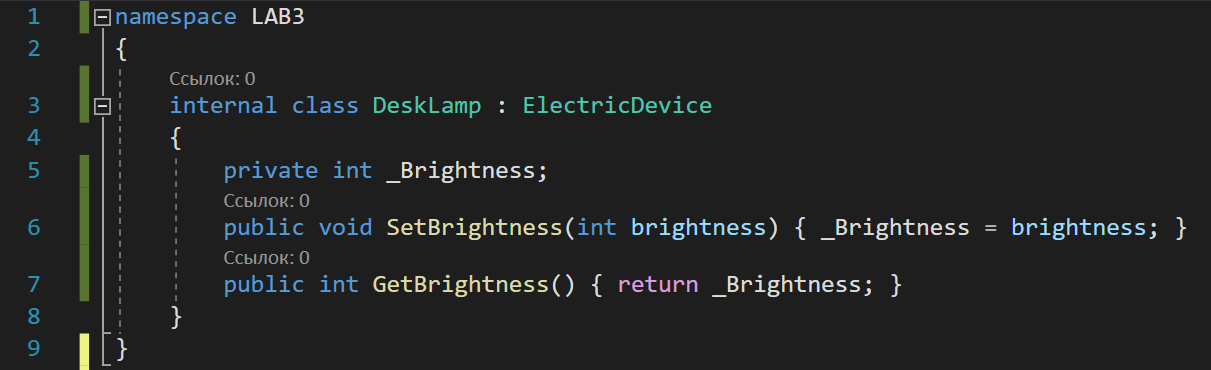
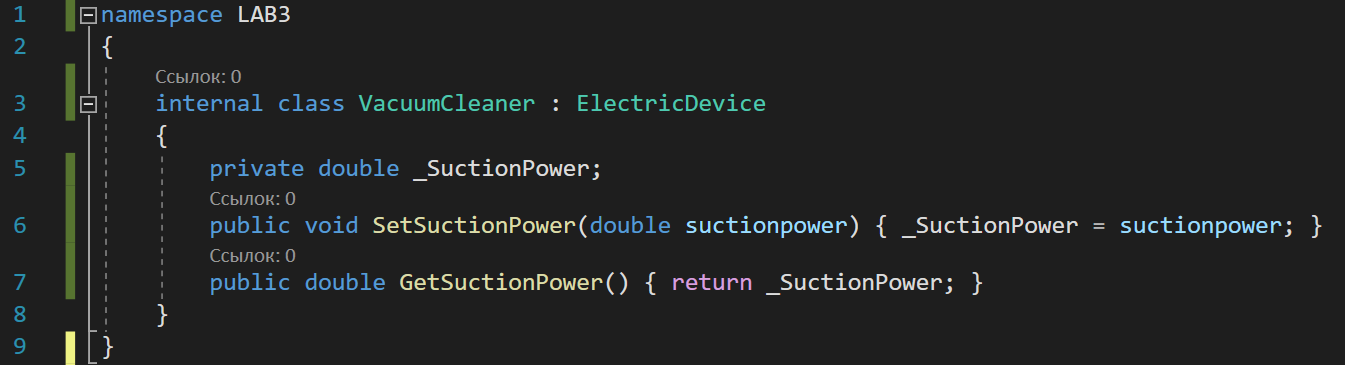


Рисунок 5 – Соединяем созданные элементы.

Рисунок 6 - Создаем библиотеку классов в MS C#







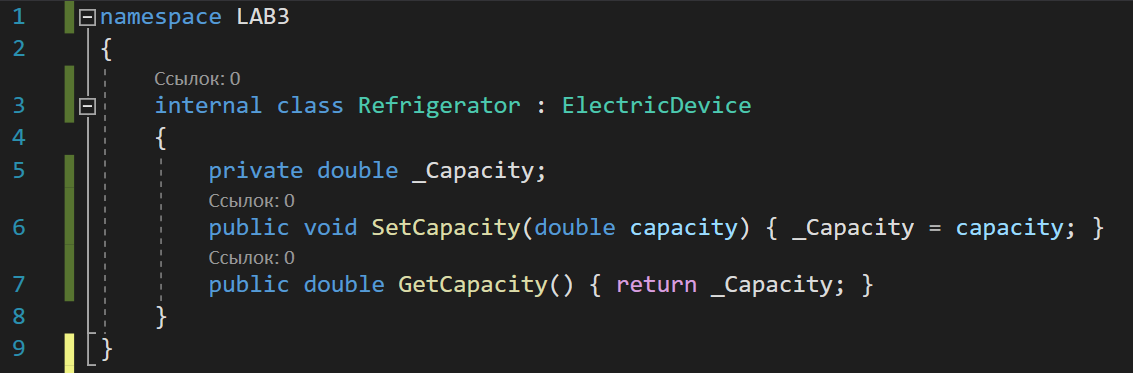


Рисунок 7,8,9,10 – Реализуем иерархию классов в MS C#

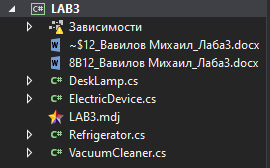


Рисунок 11 – Выносим классы в отдельные файлы.

**Вывод:** в результате лабораторной работы мыполучить опыт практической работы в создании иерархий классов предметной области с помощью UML-диаграмм. Также была описана иерархия классов предметной области в виде UML диаграммы и классов на языке C# в соответствии с выбранным вариантом задания.

**Контрольные вопросы**

1. Что такое наследование?

**Наследование** — это процесс, посредством которого один объект может приобретать свойства другого. Точнее, объект может наследовать основные свойства другого объекта и добавлять к ним свойства и методы, характерные только для него.

Наследование бывает двух видов:

* одиночное — класс (он же подкласс) имеет один и только один суперкласс (предок);
* множественное — класс может иметь любое количество предков (в Java запрещено).

1. Что такое инкапсуляция?

**Инкапсуляция** (encapsulation) – принцип, объединяющий данные и код, манипулирующий этими данными, а также защищающий в первую очередь данные от прямого внешнего доступа и неправильного использования. Другими словами, доступ к данным класса возможен только посредством методов этого же класса.

1. Что такое полиморфизм?

**Полиморфизм** (polymorphism) – механизм, использующий одно и то же имя метода для решения двух или более похожих, но несколько отличающихся задач. Целью полиморфизма применительно к ООП является использование одного имени для задания общих для класса действий. В более общем смысле концепцией полиморфизма является идея «один интерфейс, множество методов».

1. Что такое абстрактный класс?

**Абстрактный класс** – это класс, который не может быть создан непосредственно, а предназначен для того, чтобы быть наследованным другими классами. Он служит в качестве базового класса для других классов и определяет общие свойства и методы, которые могут быть унаследованы дочерними классами. Абстрактный класс содержит один или несколько абстрактных методов. Абстрактные классы используются для упрощения разработки и повторного использования кода, так как эти классы могут содержать некоторые общие свойства и методы для группы классов, которые наследуют их.