Практическое занятие № 6

Тема практического занятия: Разработка простого приложения с использованием класса.

Цель практического занятия: освоение создания и использования классов для разработки простых приложений, освоения основ объектно-ориентированного программирования и применения его принципов при разработке более сложных программных систем.

В результате выполнения данной работы обучающийся должен уметь:

1. Разрабатывать простые приложения, используя классы для организации кода и данных в качестве моделей или компонентов приложения.
2. Создавать классы с атрибутами (полями) и методами, которые определяют состояние и поведение объектов, и использовать их для решения конкретных задач и функций в приложении.
3. Проектировать и реализовывать взаимодействие между классами и объектами, включая передачу данных, вызов методов и управление состоянием объектов

знать:

1. Принципы объектно-ориентированного программирования и его основные концепции
2. Преимущества использования классов для создания модульного, масштабируемого и поддерживаемого кода.

Перечень оборудования, необходимого для выполнения задания:

- Автоматизированные рабочие места по количеству обучающихся (процессор Intel Core i7 или аналогичный, БП 700 Вт, 32 Гб ОЗУ, SSD 512 Гб, HDD 4 ТБ SATA 7200 rpm, RTX 3060 12GB);
- Автоматизированное рабочее место преподавателя (процессор Intel Core i7 или аналогичный, БП 700 Вт, 32 Гб ОЗУ, SSD 512 Гб, HDD 4 ТБ SATA 7200 rpm, RTX 3060 12GB);
- Монитор 34”, изогнутый, 3440x1440, 6 ms, 178°/178°, 300 cd/m2, 20M:1, HDMI, DP, USB, регулировка по высоте (1 монитор на одно рабочее место).
- Интерактивная панель 86” с OPS ПК

Программное обеспечение:

- Microsoft Visual Studio,

Общие теоретические сведения *:*

Использование классов для создания модульного, масштабируемого и поддерживаемого кода имеет следующие преимущества:

1. Инкапсуляция: Классы позволяют объединить данные и методы, которые работают с этими данными, в одном месте. Это позволяет представить код в виде модулей, которые легко понять и использовать другими разработчиками.

2. Абстракция: Классы позволяют представить сложные объекты или системы в виде более простых абстракций. Это делает код более понятным и удобным для сопровождения, так как сложные детали реализации скрываются за более простыми интерфейсами.

3. Наследование: Классы позволяют создать иерархию, где один класс наследует свойства и методы от другого класса. Это позволяет повторно использовать код и создавать специализированные версии классов без необходимости повторной реализации общих функциональностей.

4. Полиморфизм: Полиморфизм позволяет работать с объектами разных классов, используя общий интерфейс. Это делает код гибким и реализует принцип "одинаковое поведение для разных типов данных".

5. Модульность: Классы позволяют разделить код на отдельные модули, каждый из которых выполняет определенные функции или решает определенную задачу. Это упрощает разработку, тестирование и отладку, а также позволяет параллельно работать над разными частями проекта.

6. Расширяемость: Классы позволяют добавлять новую функциональность или изменять существующую без необходимости изменения остального кода. Это делает код более гибким и позволяет легко адаптировать его под новые требования или сценарии использования.

7. Тестирование: Использование классов позволяет легче тестировать отдельные части программы. Это способствует созданию надежного и стабильного кода.

В целом, использование классов в программировании имеет множество преимуществ, таких как читаемость, удобство сопровождения, повторное использование кода и гибкость. Классы помогают разбить сложные системы на более мелкие и понятные части, что помогает в создании качественного и эффективного кода.

Задание:

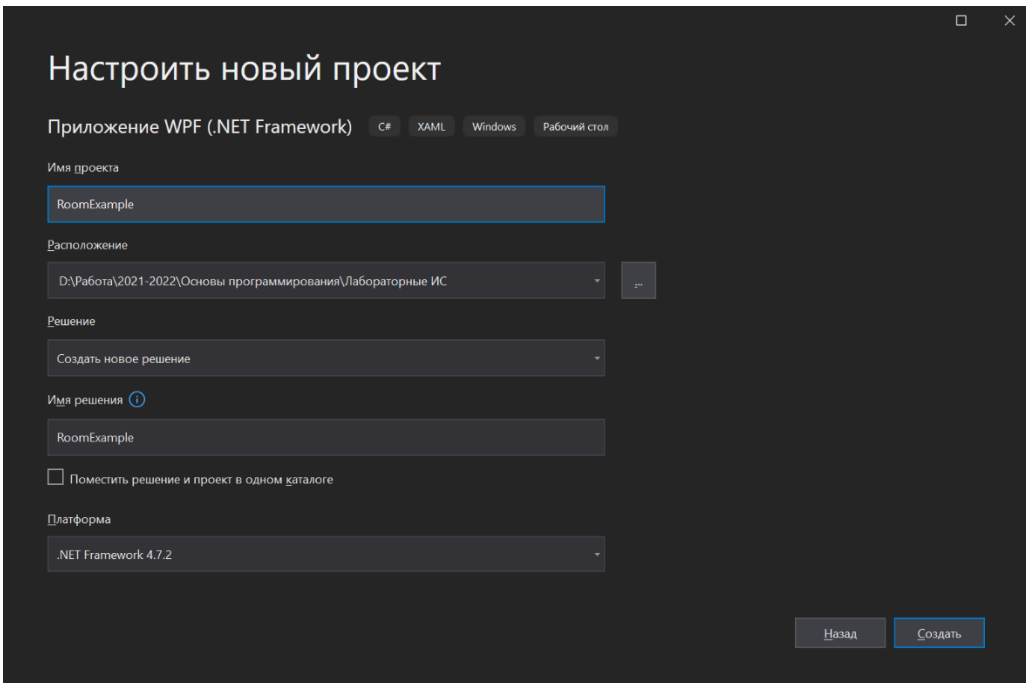
1. Используя библиотеку классов, разработанную ранее во время практического занятия №5, напишите приложение, которое реализует путешествие туриста. Задаются данные о городе и о туристе (имя, сумма денег, сколько достопримечательностей посещает за день). Есть кнопка «Новый город», после нажатия которой запоминаются поля, введенные ранее пользователем, и выводятся значения вычисляемых данных. Есть кнопка «Посетить», после нажатия которой с туриста снимается требуемая сумма денег. Эта кнопка активна только в том случае, если у туриста достаточно денег для посещения.
2. Добавьте кнопку «Статистика», после нажатия которой будет показана средняя сумма денег, которую турист тратил в городах.

Указания по технике безопасности:

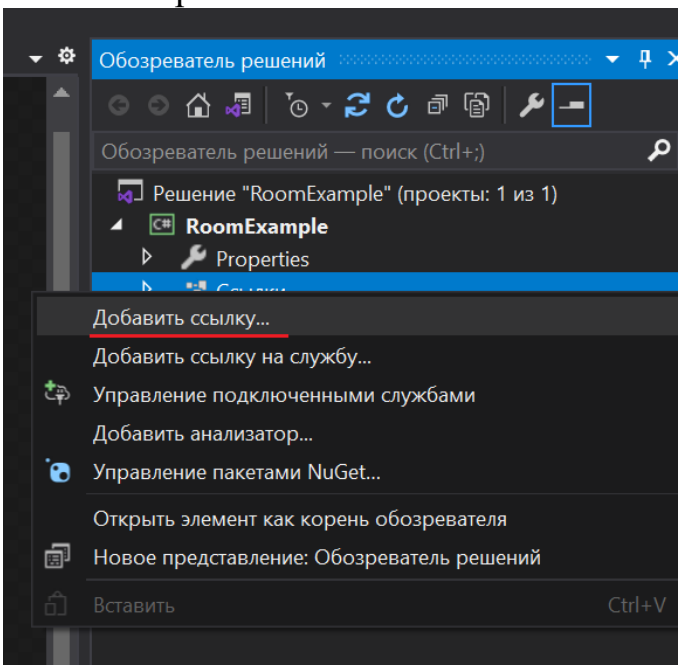
Инструкция по технике безопасности при работе в лаборатории, оборудованной компьютерной техникой.

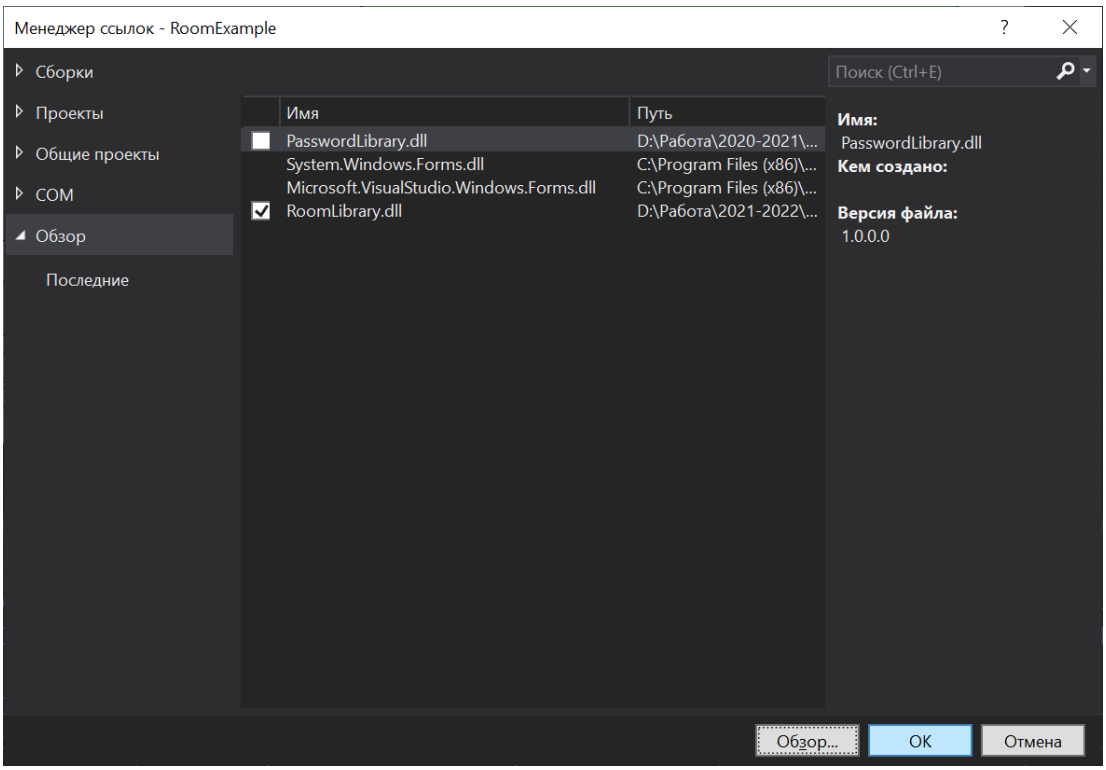
Технология выполнения работы (*этапы, последовательность действий*):

1. Создайте проект для приложения.



2. Подключите к проекту библиотеку классов. Для этого добавьте библиотеку в ссылки проекта:

13. 

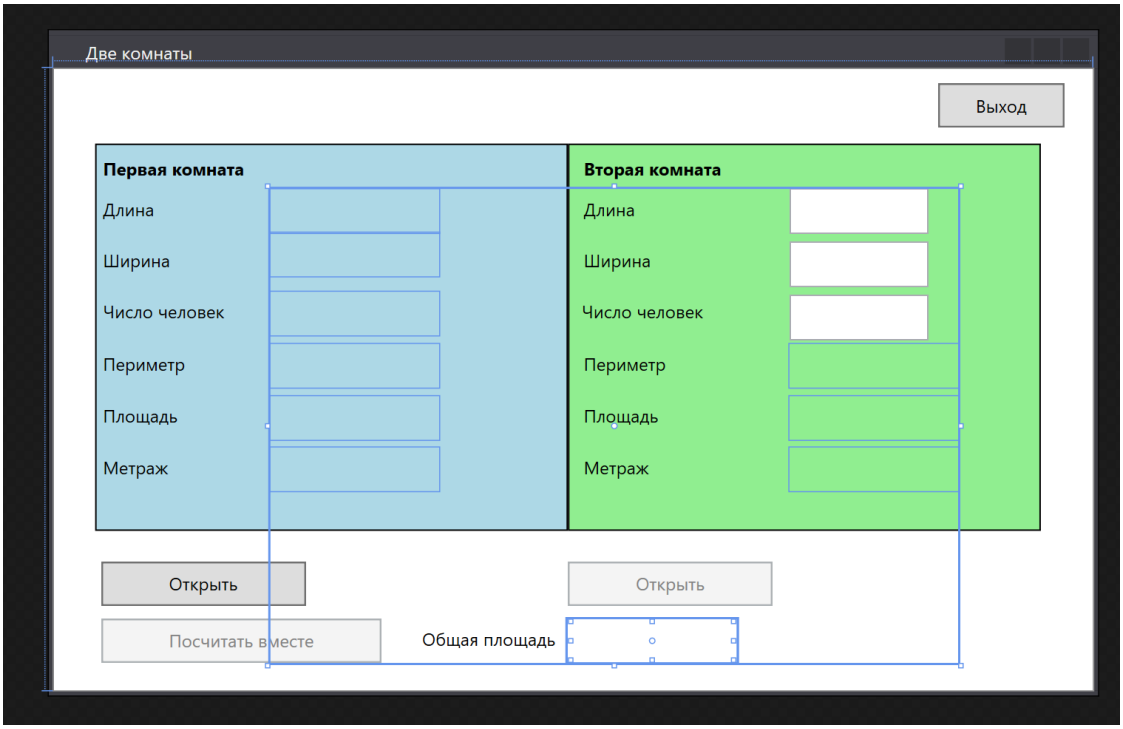


А также добавьте в коде соответствующее пространство имен:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using RoomLibrary;

namespace RoomExample
{
    /// <summary>
    /// Логика взаимодействия для MainWindow.xaml
    /// </summary>
    Ссылок: 2
    public partial class MainWindow : Window
    {
        Ссылок: 0
        public MainWindow()
        {
            InitializeComponent();
        }
    }
}
```

3. Разработайте интерфейс приложения в соответствии с заданием. Пример интерфейса для другого приложения:



4. Реализуйте обработчик события кнопки «Новый город», проверьте ее работоспособность.
5. Реализуйте обработчик кнопки «Посетить», проверьте ее работоспособность.
6. Реализуйте обработчик кнопки «Статистика», проверьте ее работоспособность.
7. Учтите формат выдаваемых данных: у вещественных значений должно выполняться округление с точностью до двух знаков после запятой.

Требование к отчету:

1. Подключена библиотека классов.
2. Приложение компилируется, запускается.
3. Есть возможность ввести данные о туристе.
4. Кнопка «Новый город» работает корректно: после ее нажатия обеспечивается запоминание полей и вычисление данных о городе.
5. Кнопка «Посетить» работает корректно: у туриста уменьшается сумма денег на сумму, соответствующую стоимости проживания в городе за требуемое для его посещения количество дней.
6. Если у туриста не хватает денег для посещения данного города, кнопка «Посетить» не должна быть активной.
7. Кнопка «Статистика» работает корректно.
8. Все вещественные данные выдаются с точностью до двух знаков после запятой.

Контрольные вопросы:

1. Расскажите о преимуществах объектно-ориентированного программирования и использования классов.
2. Как подключить библиотеку классов к приложению?
3. Что такое обработчик события?
4. С помощью чего инициализируется экземпляр класса?
5. Каким образом можно выполнить округление с заданной точностью?
6. Как сделать кнопку неактивной? Как потом снова ее активизировать?

Основные и дополнительные источники, электронные ресурсы:

1. Подбельский, В. В. Язык C#. Базовый курс: учебное пособие / В. В. Подбельский. - 2-е изд., перераб. и доп. - Москва: Финансы и статистика, 2022. - 408 с. - ISBN 978-5-00184-079-4. - Текст: электронный. - URL: https://znanium.com/catalog/product/1913989
2. Гуриков, С. Р. Основы алгоритмизации и программирования на Python: учебное пособие / С.Р. Гуриков. — Москва: ИНФРА-М, 2023. — 343 с. — (Среднее профессиональное образование). - ISBN 978-5-16-016906-4. - Текст: электронный. - URL: https://znanium.com/catalog/product/1927269.