Практическое занятие № 4

Тема практического занятия: Описание классов.

Цель практического занятия: овладение навыками проектирования и описания классов.

В результате выполнения данной работы обучающийся должен уметь:

1. Создавать библиотеку классов.
2. Создавать класс, описывать поля и свойства.

знать:

1. Синтаксис описания класса.
2. Модификаторы доступа к элементам класса.

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:

Класс в объектно-ориентированном программировании - это шаблон или описание, определяющее структуру и поведение объектов, которые могут быть созданы на основе этого класса. Класс определяет состояние и методы данных, которыми обладает объект.

Он определяет набор переменных, известных как поля класса, которые представляют состояние объекта, а также методы, которые определяют действия (поведение), которые объект может выполнять.

Классы в объектно-ориентированном программировании обеспечивают понятия инкапсуляции и абстракции. Они позволяют группировать данные и функциональность в один объект, что способствует повторному использованию кода, упрощает его понимание и поддержку.

Когда объект создается на основе класса, мы называем этот объект экземпляром класса. Каждый экземпляр получает собственные копии полей класса и может использовать методы класса для выполнения определенных действий.

Классы являются основными строительными блоками в объектно-ориентированном программировании и играют ключевую роль при организации и структурировании кода.

В объектно-ориентированном программировании класс состоит из нескольких элементов, таких как поля, свойства и методы. Вот их краткое описание:

1. Поля (переменные класса): это переменные, которые определены внутри класса и хранят данные, относящиеся к объектам этого класса. Поля могут быть публичными, приватными или защищенными, и их значение может быть изменено или получено через методы класса.

2. Свойства: это специальные методы, которые предоставляют доступ к частным полям класса. Свойства позволяют контролировать доступ к данным и предоставлять дополнительную логику при их чтении или записи. Свойства могут быть только для чтения (только get), только для записи (только set) или для чтения и записи (get и set).

3. Методы: это функции, определенные внутри класса, которые определяют поведение объектов этого класса. Методы могут выполнять операции над данными класса, взаимодействовать с другими объектами или возвращать результаты вычислений. Методы могут быть публичными, приватными или защищенными, в зависимости от того, какой уровень доступа должен быть предоставлен другим классам.

Поля, свойства и методы взаимодействуют внутри класса, их комбинация определяет состояние и поведение объектов, созданных на основе этого класса. Поля хранят данные объекта, свойства обеспечивают доступ к этим данным с дополнительной логикой, а методы выполняют действия и операции, связанные с этими данными.

Элементы класса могут иметь разный уровень доступности в зависимости от модификаторов доступа, указанных при их описании. Модификаторы доступа в объектно-ориентированном программировании определяют уровень доступности элементов класса (полей, свойств и методов) для других классов и объектов. Существует несколько модификаторов доступа:

1. Публичный (public): элементы, объявленные с модификатором public, доступны из любого места программы. Они могут быть обращены и изменены из любого класса или объекта.

2. Приватный (private): элементы, объявленные с модификатором private, доступны только внутри класса, в котором они были объявлены. Они не могут быть обращены или изменены из других классов или объектов. Приватные элементы используются для скрытия реализации и предоставления контролируемого доступа к данным.

3. Защищенный (protected): элементы, объявленные с модификатором protected, доступны внутри класса, в котором они были объявлены, а также в подклассах (наследниках) этого класса. Они не могут быть обращены из

других классов или объектов. Защищенные элементы используются для наследования и предоставления доступа подклассам к родительским данным.

4. Внутренний (internal): элементы, объявленные с модификатором internal, доступны только внутри сборки (библиотеки или проекта). Они не могут быть обращены из других сборок.

Корректный выбор модификаторов доступа важен для создания хорошо структурированного и безопасного кода.

Задание:

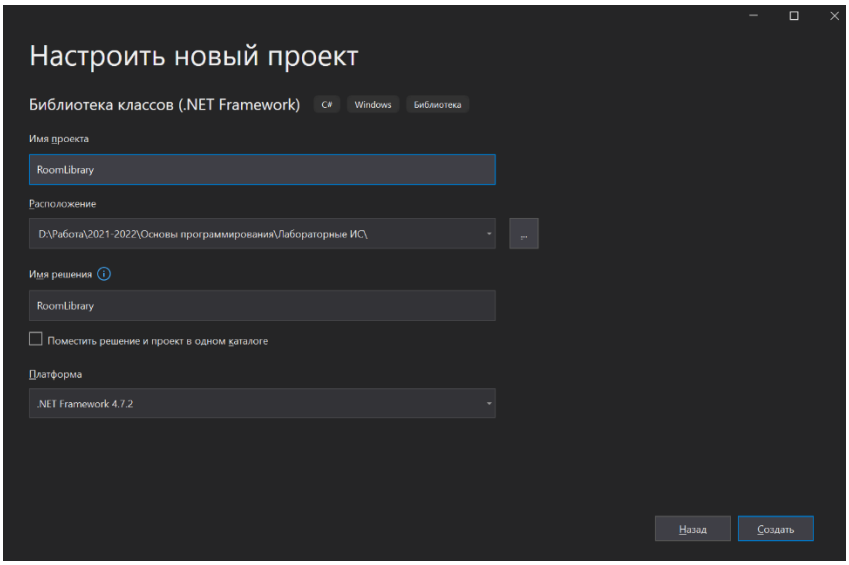
1. Опишите класс «Город», который хранит информацию о городе, который собирается посетить путешественник: название, численность населения, площадь, число достопримечательностей, стоимость проживания за день.
2. Для каждого из полей опишите соответствующее полю свойство. В свойствах предусмотрите защиту от ввода некорректных данных: название города не должно быть пустой строкой, численность населения и число достопримечательностей могут быть только целыми положительными числами, площадь и стоимость проживания – положительными числами. Если данные некорректны, им должно быть присвоено некоторое корректное значение.
3. Разработайте приложение, в котором будет объявлен экземпляр класса «Город». Реализуйте возможность заполнения полей и определения их значений.

Указания по технике безопасности:

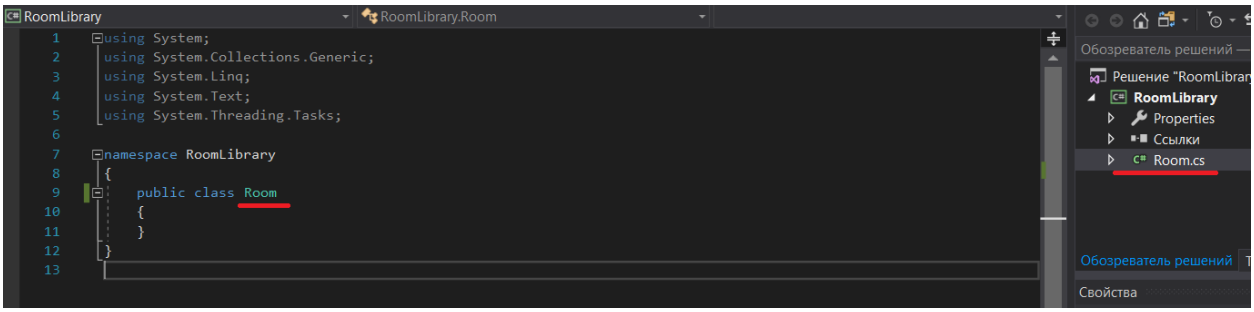
Инструкция по технике безопасности при работе в лаборатории, оборудованной компьютерной техникой.

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

1. Создайте проект «библиотека классов» или его аналог.



2. Создайте соответствующий заданию класс (или переименуйте созданный по умолчанию). Здесь и далее для примера будет рассматриваться другой класс, задание для класса «Город» выполните по аналогии.



3. Опишите поля класса. Пример описания полей класса:

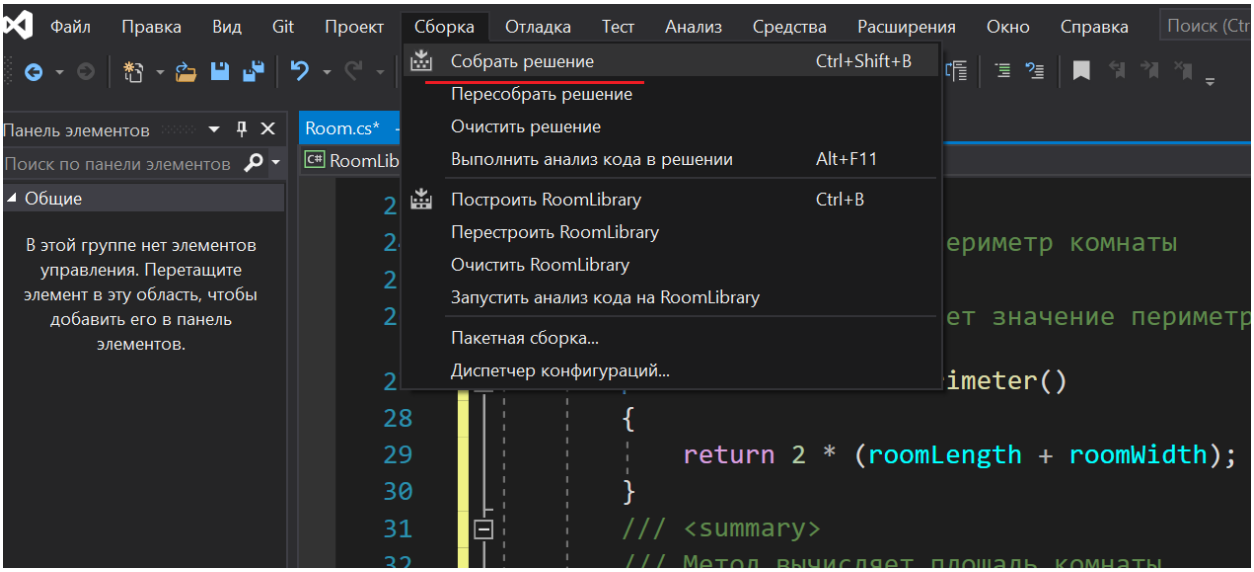
```
namespace RoomLibrary
{
    Ссылок: 0
    public class Room
    {
        double roomLength;  //длина комнаты
        double roomWidth;   //ширина комнаты
    }
}
```

4. Опишите соответствующие полям свойства. Предусмотрите в них логику, которая не будет позволять задавать полям некорректные значения.
Пример описания свойств, соответствующих полям:

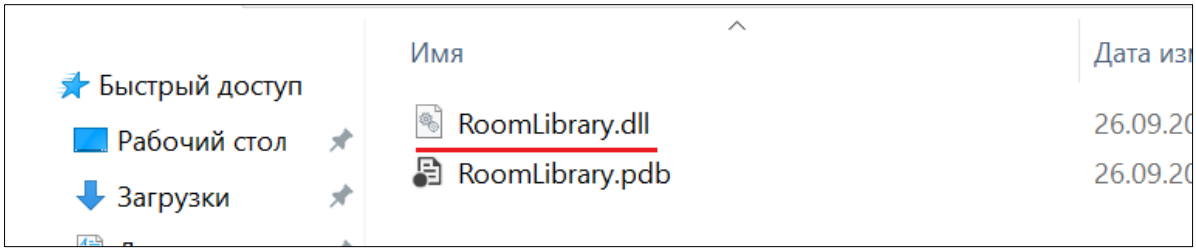
```
Ссылок: 0
public class Room
{
    double roomLength;  //длина комнаты
    double roomWidth;   //ширина комнаты
    Ссылок: 0
    public double RoomLength
    {
        get { return roomLength; }
        set { roomLength = value; }
    }
    Ссылок: 0
    public double RoomWidth
    {
        get { return roomWidth; }
        set { roomWidth = value; }
    }

}
```

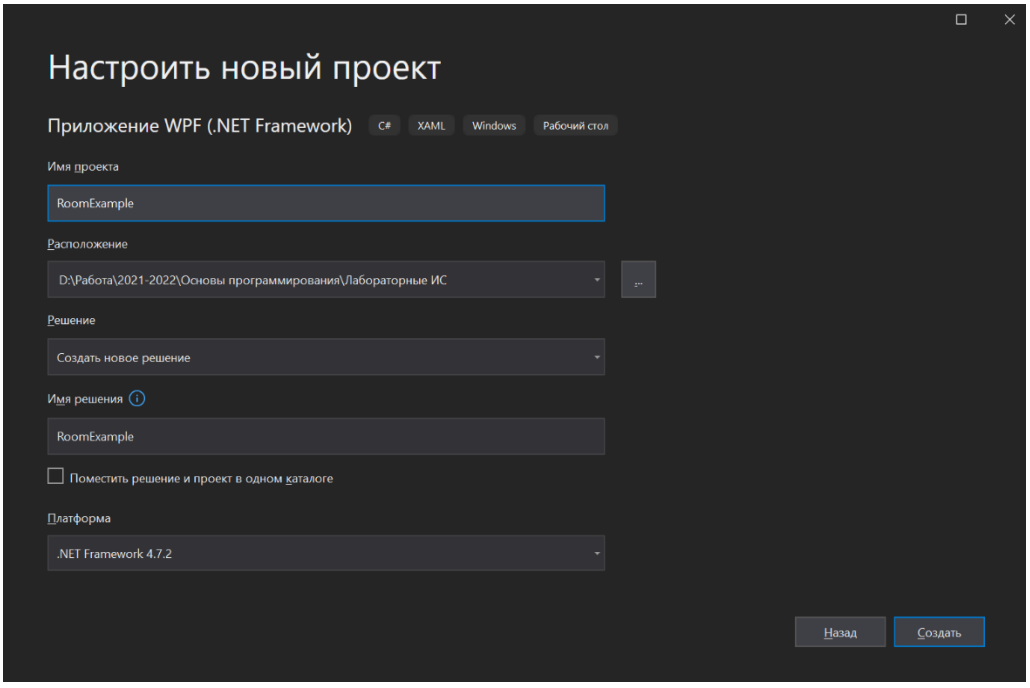
5. Выполните сборку библиотеки классов. Для этого в меню «Сборка» выберите пункт «Собрать решение»:



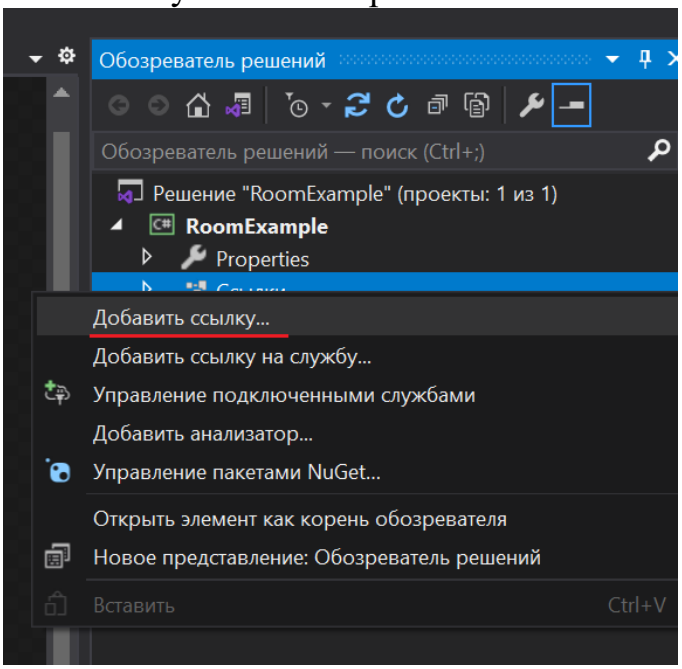
6. Убедитесь, что она прошла успешна, и что в проект добавлена библиотека (файл с расширением .dll).

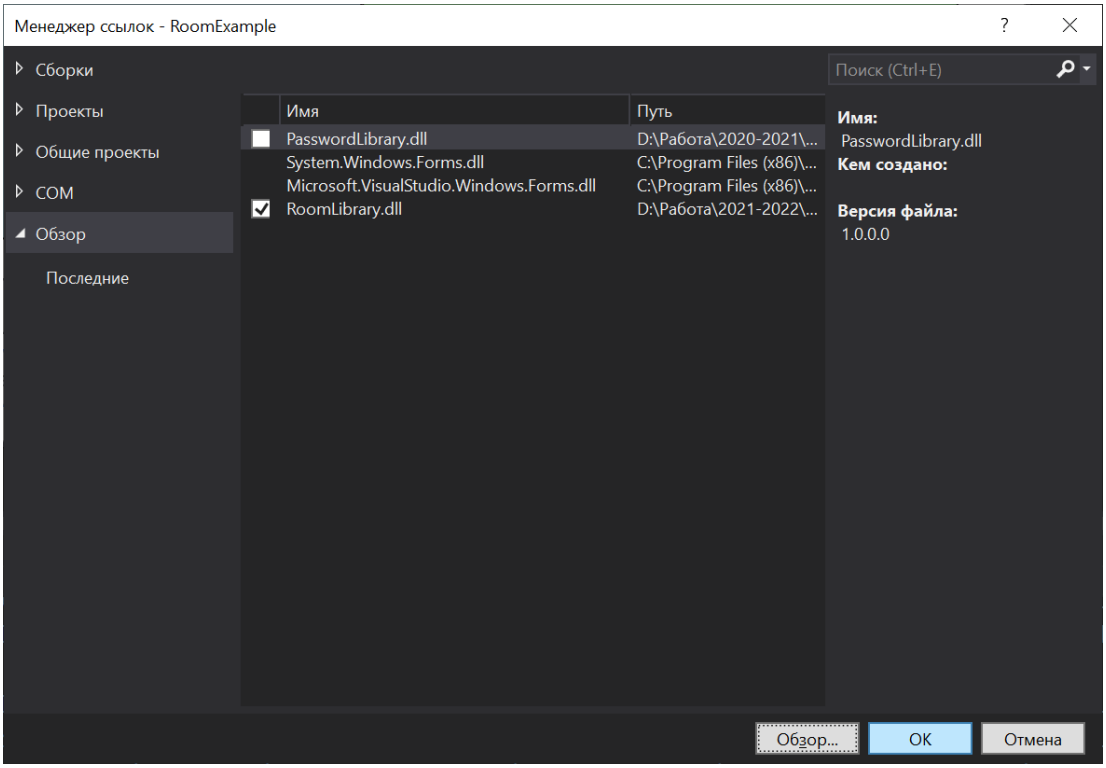


7. Создайте проект для приложения.



8. Подключите к нему свою библиотеку классов. Для этого добавьте библиотеку в ссылки проекта:



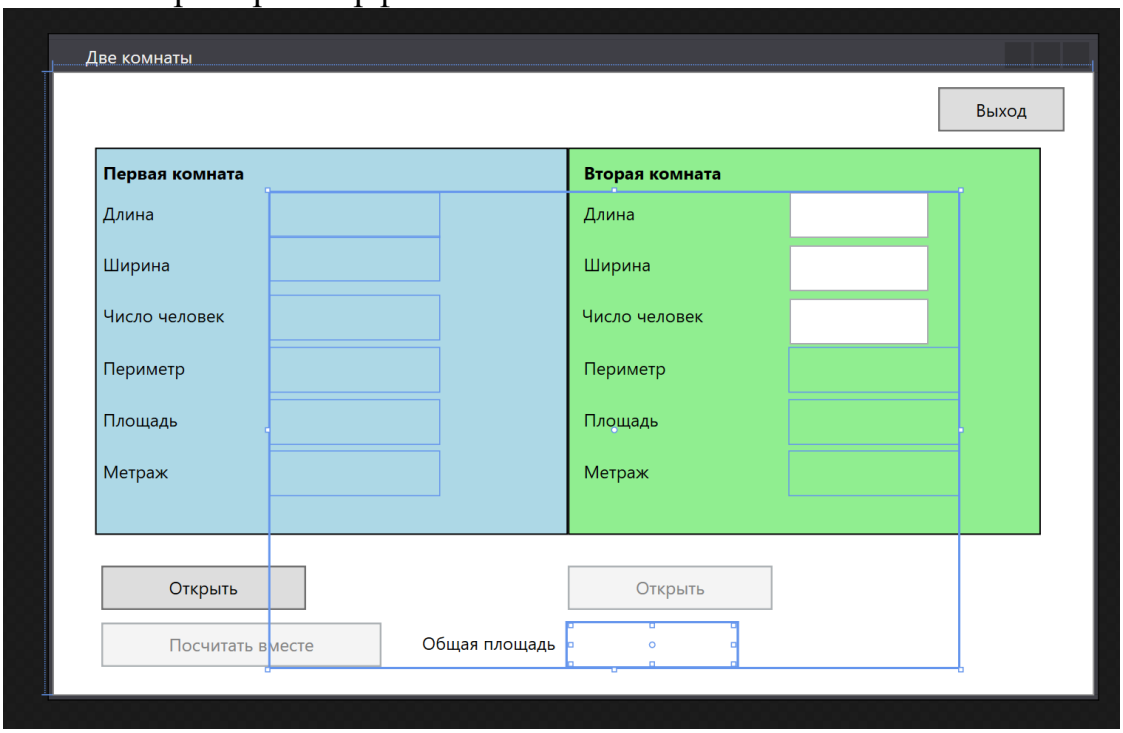


А также добавьте в коде соответствующее пространство имен:

```
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();
        }
    }
}
```

9. Разработайте интерфейс взаимодействия с пользователем: текстовые поля для значений, метку для выдачи информации, кнопку для инициализации объекта. Пример интерфейса



10. Обеспечьте возможность пользователю инициализировать объект «Город», введя значения. Значения должны записываться в поля класса.
11. Добавьте в обработчик события кнопки выдачу полной информации об объекте.
12. Убедитесь в корректности работы приложения, в том числе и в случае ввода недопустимых данных (нецелое число достопримечательностей, отрицательная площадь и др.).

Требование к отчету:

1. Описан класс «Город», имеющий поля и свойства.
2. Принцип инкапсуляции при описании класса не нарушен.
3. Типы данных для полей выбраны корректно.
4. Скомпилирована библиотека классов (файл .dll).
5. Разработано приложение, которое позволяет вводить значения полей города и заполнять их.
6. Приложение позволяет получить информацию о городе.
7. Приложение корректно работает даже при недопустимых значениях входных данных.

Контрольные вопросы:

1. Объясните, для чего необходимо описывать дублирующие поля свойства класса?
2. Что означает модификатор private?
3. Что означает модификатор public?
4. Что означает модификатор protected?
5. Что означает модификатор internal?
6. Что обозначают блоки get и set в свойствах?
7. Что такое файл с расширением .dll?
8. После каких действий появляется файл с расширением .dll?
9. Что необходимо сделать, чтобы можно было пользоваться классами библиотеки в своем приложении?
10. Для чего необходимо вызывать конструктор?
11. Что такое обработчик события?

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.