Министерство образования Московской области

Государственное образовательное учреждение высшего образования Московской области

«Государственный гуманитарно-технологический университет»

**Ликино-Дулевский политехнический колледж – филиал ГГТУ**

**Отчёт по практической работе №13**

по дисциплине

«МДК 01.01. Разработка программных модулей»

**Выполнил(а):**

Немтырёва Ксения Антоновна

студент(ка) группы

ИСП.20А

09.02.07 Информационные системы и программирование

очной формы обучения

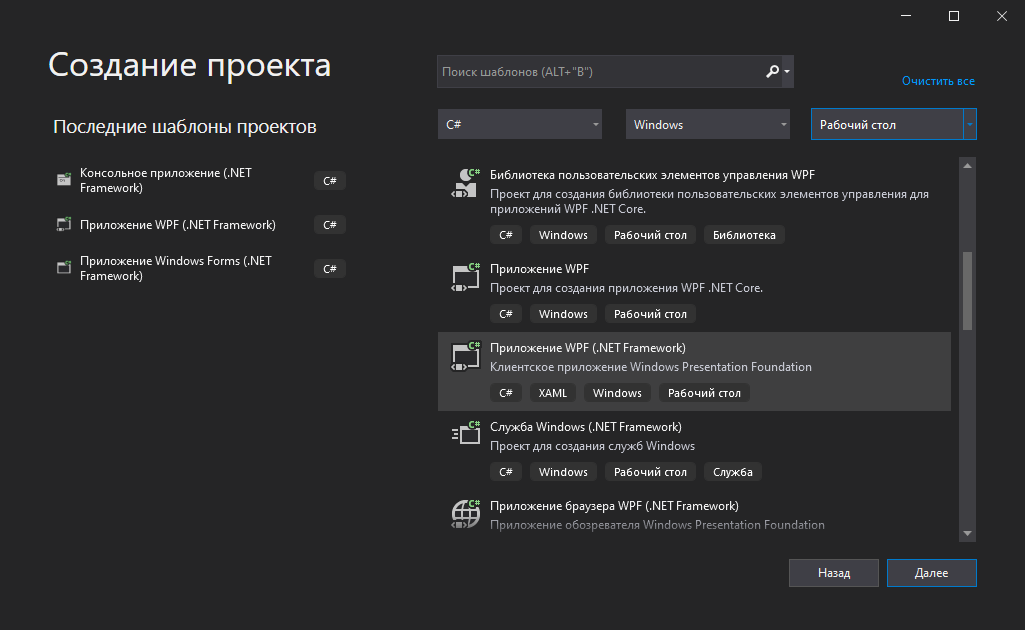
Проверил:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

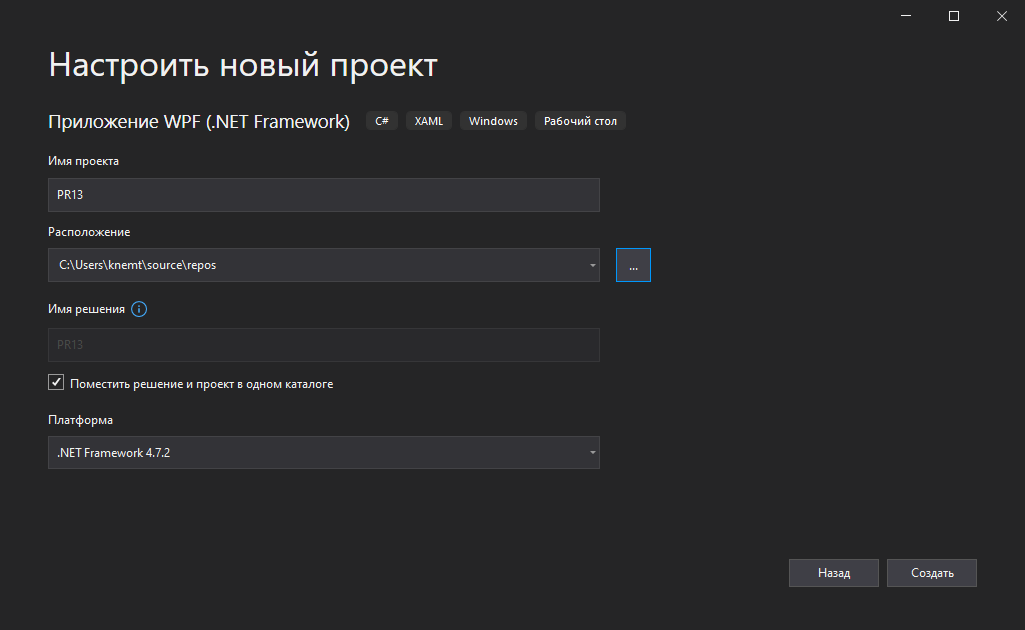
Оценка

\_\_\_\_\_\_\_\_\_/\_\_ \_\_\_\_\_\_\_\_\_\_/

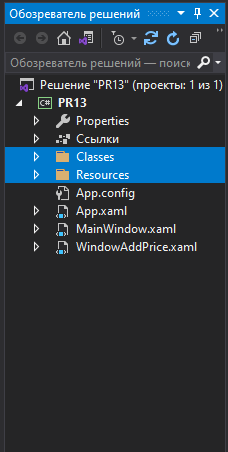
1. Создаём проект «Приложение WPF»



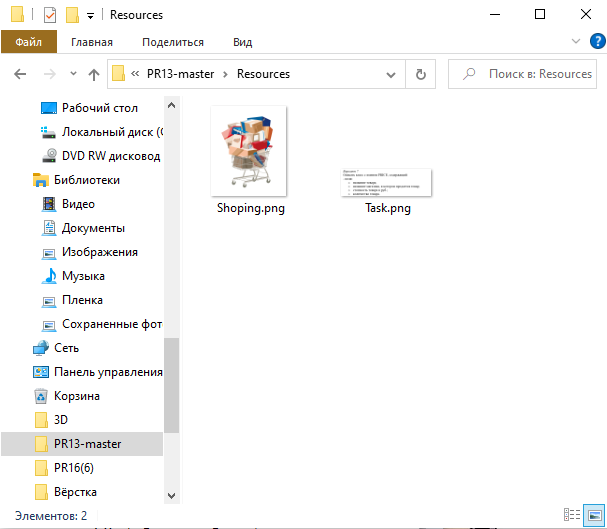
1. Называем, указываем расположение. Нажимаем «Создать».



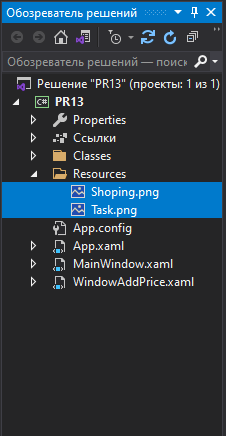
1. Создаём структуру проекта (Classes, Resources).



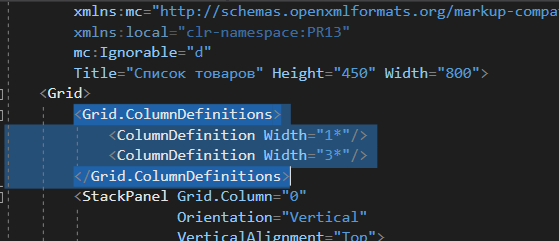
1. Делаем скриншот задания, сохраняем в папку с проектом под названием «Task».



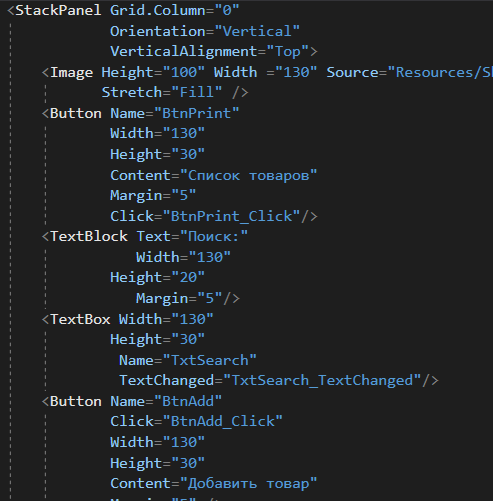
1. Перетаскиваем в проект в папку «Resources». Таким же способом получаем и другие картинки для проекта.



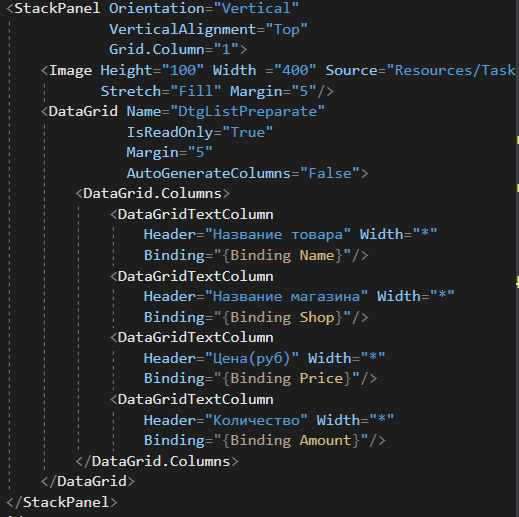
1. Делаем разметку интерфейса



1. Создаём StackPanel, в нём размещаем несколько кнопок.



1. В соседнем столбце в StackPanel размещаем картинку задачи и DataGrid.



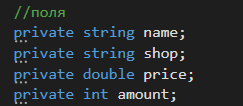
1. Меняем заголовок окна.



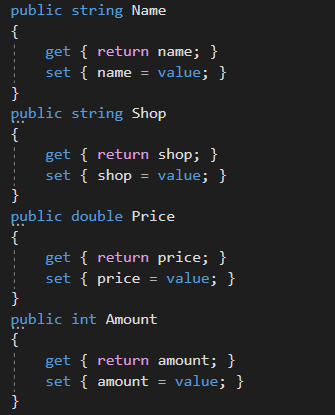
1. Переходим в папку Classes, в ней создаём класс PRICE.



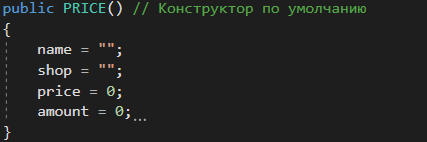
1. Создаём поля (с маленькой буквы).



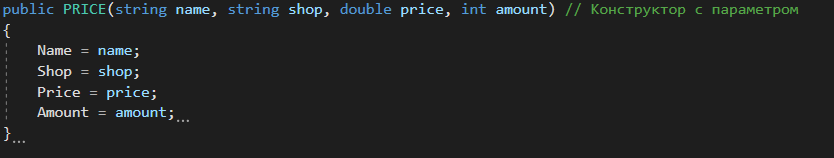
1. Прописываем свойства к полям public (с большой буквы).



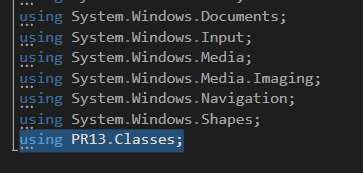
1. Создаём конструктор по умолчанию.



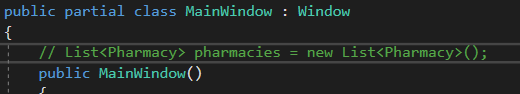
1. Конструктор с параметрами.



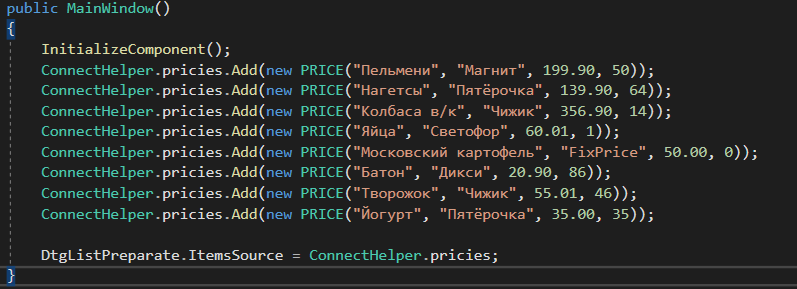
1. В MainWindows.cs подключаем наш класс.



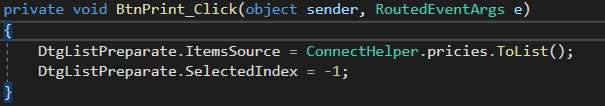
1. В блоке видимости классов окна создаём список объектов нашего класса. (Потом мы будем использовать ConnectHelper)



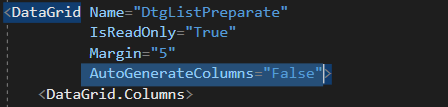
1. В блоке инициализации создать набор данных, который будет содержать данные в соответствии с полями класса, которые мы описали.



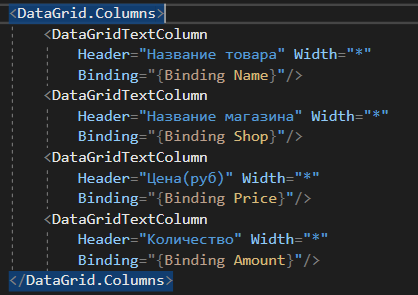
1. В определении кнопки «Список товаров» добавляем наш список.



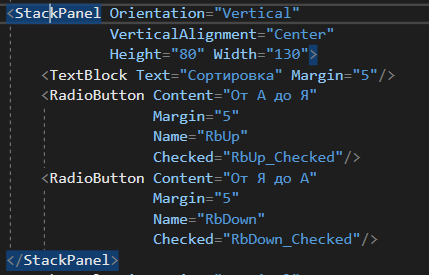
1. Идём в настройку DataGrid и отключаем автоматическую генерацию.



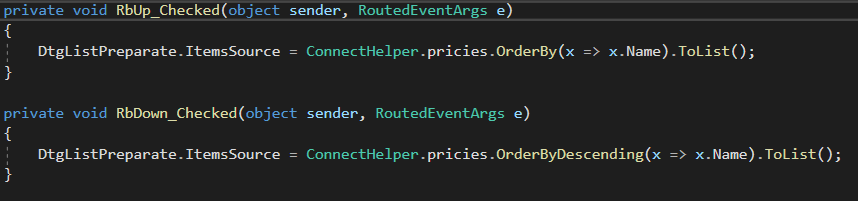
1. В DataGrid настраиваем столбцы: изменяем название столбцов, привязываем столбцы к свойствам класса, настраиваем ширину.



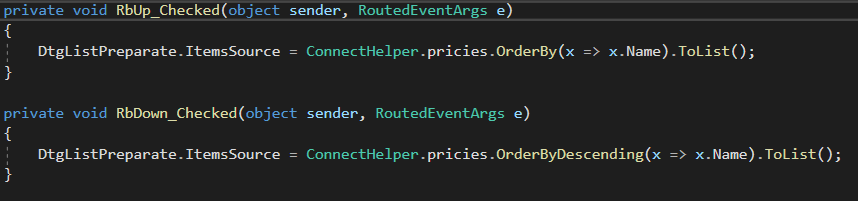
1. В первой колонке интерфейса создадим StackPanel и добавим в него TextBlock «Сортировка» и 2 RadioButton, к ним прописываем события. Они будут отвечать за фильтрацию.



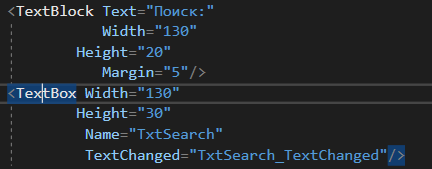
1. В определении этих событий копируем код из определения кнопки «Список товаров» вставляем в фильтрацию «От А до Я» и тут же пропишем запрос Linq (OrderBy).



1. Тоже самое и в определении фильтрации «От Я до А», только с командой OrderByDescending.



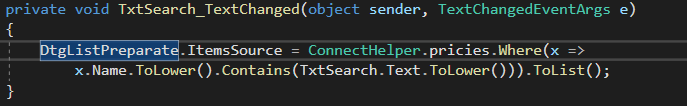
1. Вставляем TextBox. Он будет отвечать за поиск. Сверху через TextBlock прописываем «поиск».



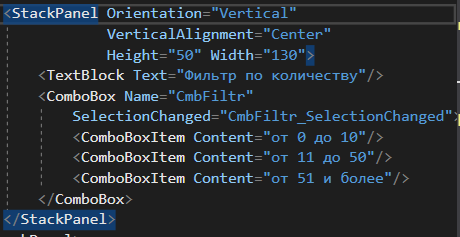
1. Добавляем событие TextChanged для текстового поля и создаём для него новый обработчик.



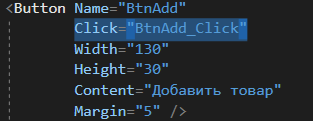
1. Прописываем запрос Linq.



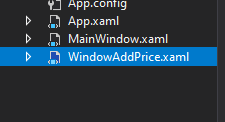
1. В интерфейсе после сортировки создаём StackPanel и размещаем туда: TextBlock «фильтр по количеству» и ComboBox, в котором будут храниться критерии фильтрации, для него же прописываем событие SelectedChanged.



1. Создаём событие для кнопки «Добавить товар» и переходим к её определению.



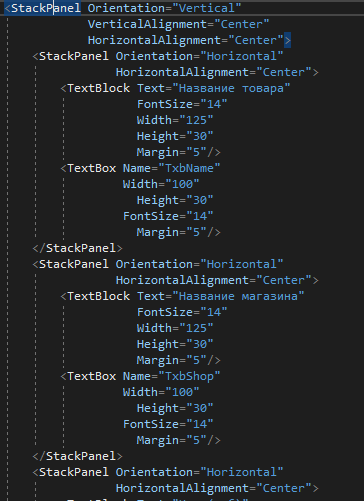
1. Добавим окно WindowsAddPrice.



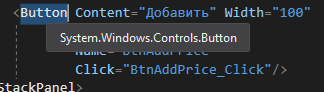
1. Поменяем заголовок окна.



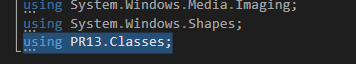
1. Размечаем окно. Создаём StackPanel, в которой будут маленькие StackPanel с TextBlock и TextBox(для каждого задаём имена).



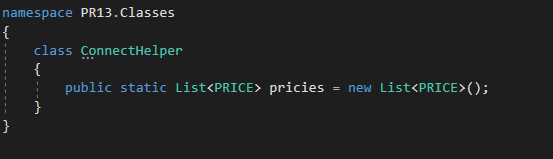
1. Вниз добавляем кнопку «Добавить», с именем и обработчиком события.



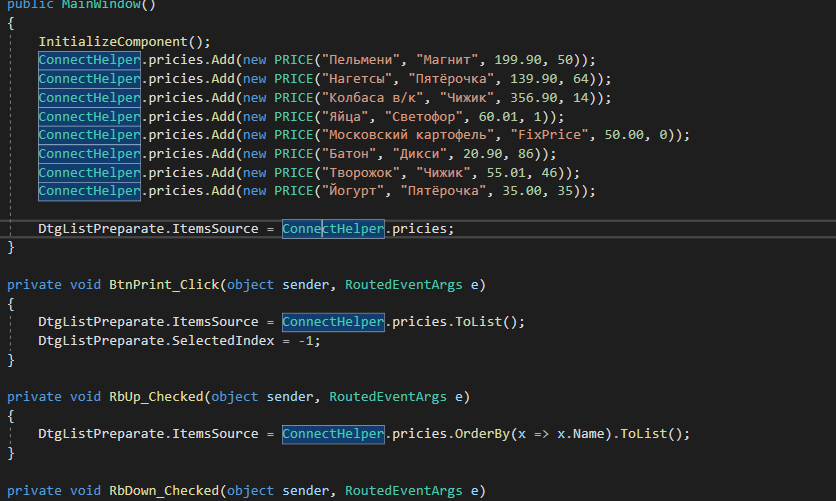
1. В коде cs прописываем using с названием проекта и папкой класса.



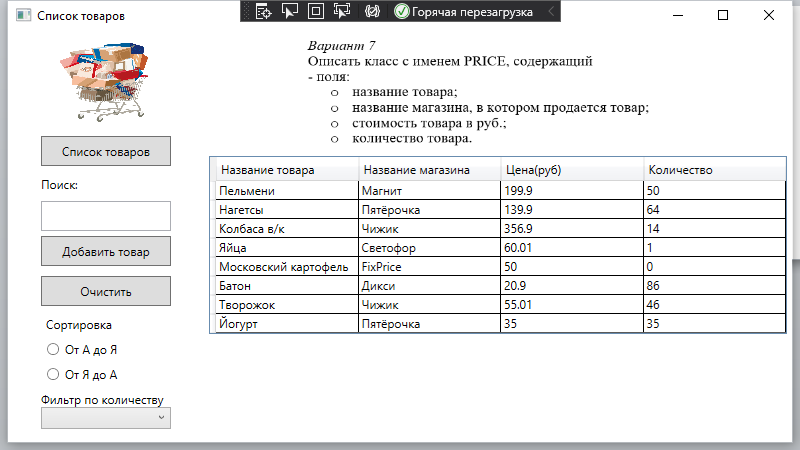
1. Чтобы наш список был виден, создаём класс помощник в соединении.



1. Используем его везде, где нужен доступ к списку.



Интерфейс главного окна:



Интерфейс окна добавления товара:

