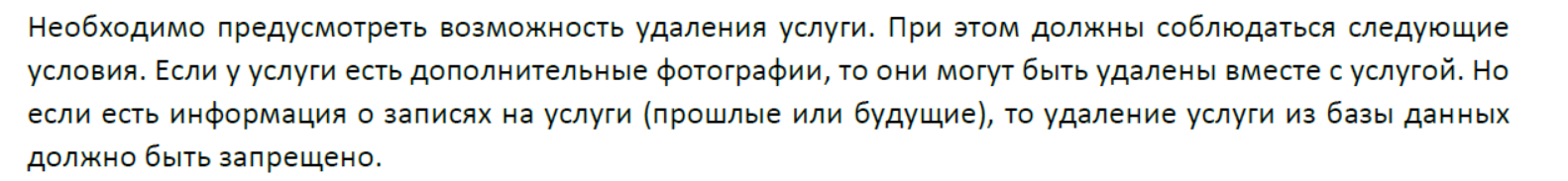
Удаление услуг. Работа с изображениями.

Постраничный вывод

**1. Клонировать с гитхаба предыдущую версию проекта (итог лабы6)**

**2. Поработаем над удалением услуг. Работаем в файле ServicePage.xaml.cs**

Задание из файла Сессия1.pdf:



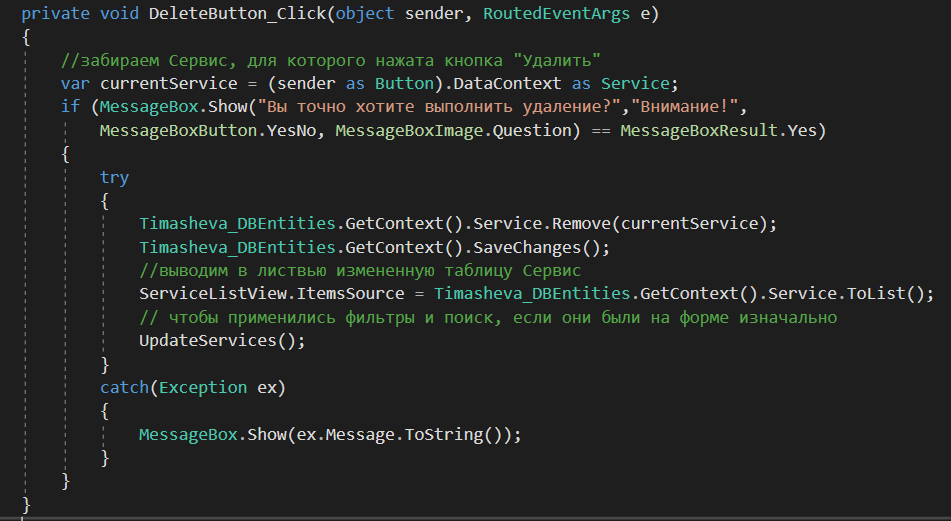
Для реализации функции удаления отдельное окно не потребуется. Это значительно сократит время на разработку, но удаление требует к себе особого внимания. Любые действия, безвозвратно изменяющие данные в базе данных, должны запрашивать подтверждение пользователя. Именно поэтому, в первую очередь, для нажатия на кнопку удаления мы реализуем сообщение с вопросом: действительно ли пользователь хочет это сделать.

В сообщении будем спрашивать пользователя: «Вы точно хотите удалить услугу?». Укажем здесь заголовок сообщения — «Внимание», затем укажем, какие кнопки доступны при диалоге с пользователем: «Да» или «Нет», и выберем изображение — «Question»:

Добавьте обработчик события для кнопки «Удалить» в ServicePage.xaml:

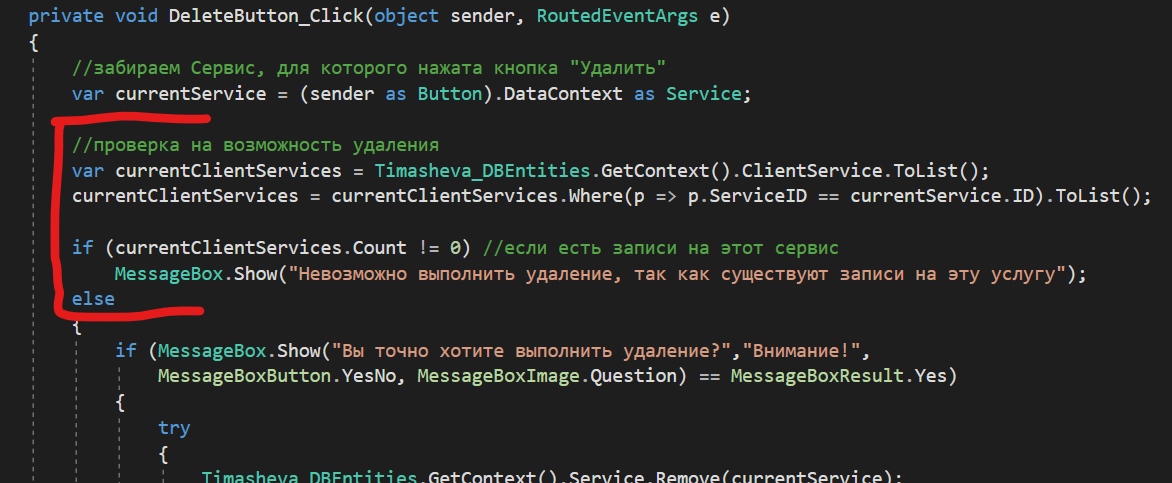
**Click="DeleteButton\_Click"**

Теперь код:



Но этот код удаляет любую услугу в базе, а по условию требуется удалять **только те услуги, на которые нет записей клиентов.**

Добавим проверку:

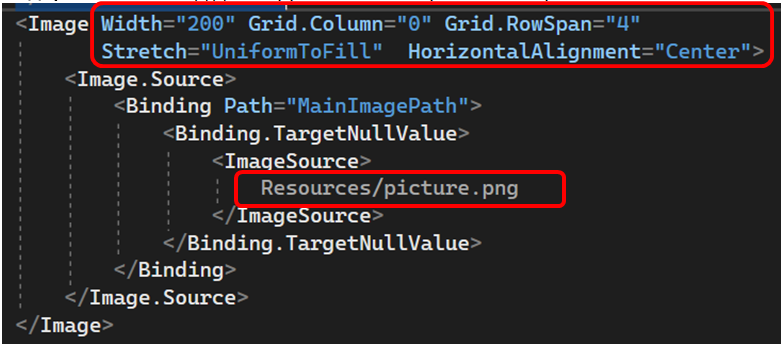


**3. Поработаем над отображением рисунков.**

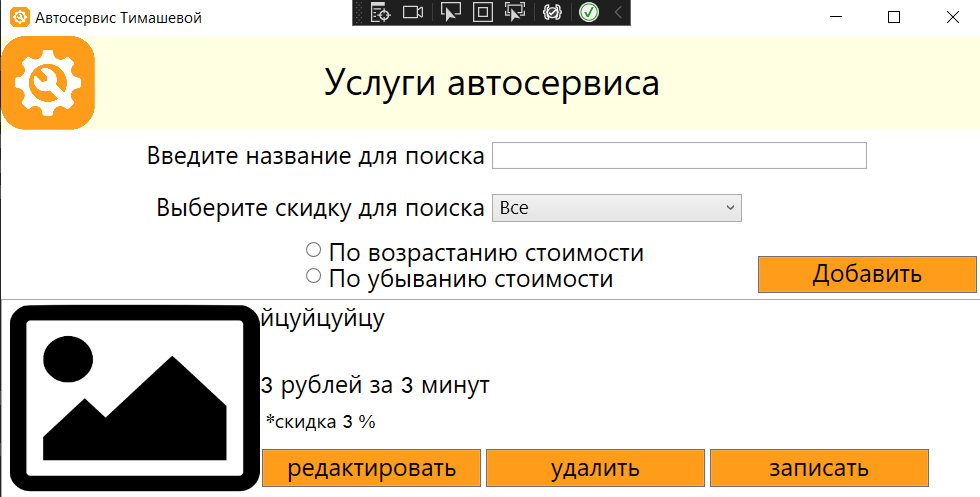
В ресурсах к заданию есть архив с фото услуг. Требуется разархивировать, добавить в проект через *Обозреватель решений* папку под названием «Услуги автосервиса», в нее поместить все картинки.

В 4 лабе уже прописывали код для отображения рисунков через привязку к полю MainImagePath, и для отображения рисунка-заглушки, если у услуги нет изображения. Нужно будет заменить рисунок-заглушку на picture.png.

**Подправим** этот код для корректного отображения картинки:

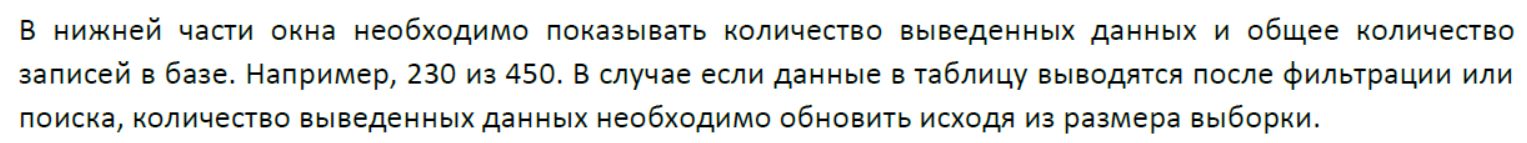


Осталось проверить работу, а именно добавить новую услугу – и так как для нее не назначено картинки, отобразится картинка-заглушка.

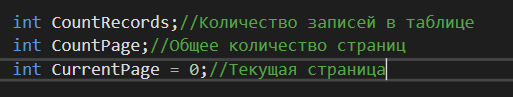


**4. Поработаем над постраничным выводом услуг.**

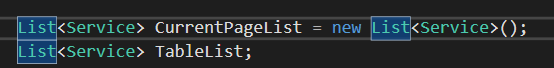
4.1. В задании сказано:



Понадобятся следующие глобальные переменные:



Кроме того еще понадобятся следующие глобальные переменные:

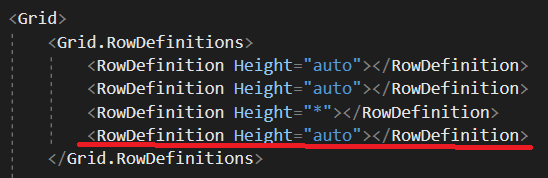


CurrentPageList – текущий лист, который заносится в таблицу;

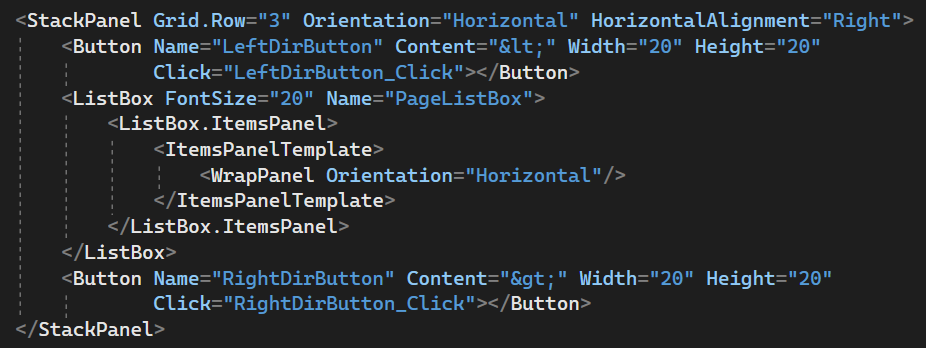
TableList – лист, содержащий все записи, все сортировки/фильтры/поиски применяются к данной переменной.

4.2.Все записи разобьем по 10 записей в одной странице, они заносятся в листвью, перебор идет через индексы, к примеру, 100 записей будет разбито на 10 страниц по 10 записей, для 1 страницы будут внесены с 0 по 9 записи, для 2 – с 10 по 19 и т.д.

Для отображения перехода по страницам используем две кнопки «<», «>» и второй листвью для списка страниц. Сначала добавим еще одну строку в грид для этих элементов:



4.3. Затем StackPanel с тремя элементами:



4.4. Для обработки перехода по страницам пропишем функцию ChangePage с двумя аргументами:

1) Параметр int direction (направление перелистывания), возможны 3 значения:

0 - начальная загрузка,

1 - предыдущая страница,

2 - следующая страница,

2) nullable целым selectedPage.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Нередко бывает удобно, чтобы переменная/параметр значимого типа могли принимать значение null, Например, получаем числовое значение из базы данных, которое в БД может отсутствовать. То есть, если значение в базе данных есть - получим число, если нет - то null.

Чтобы было возможным присвоение параметру значимого типа значения null, этот параметр должен представлять тип **nullable**. Для этого после названия типа указывается знак вопроса ?

**int? x;**

**x = null;**

Здесь переменная x представляет не просто тип int, а тип **int?** - тип, параметры которого могут принимать как значения типа int, так и значение null. В данном случае мы передаем ей значение null. Но также можно передать и значение типа int:

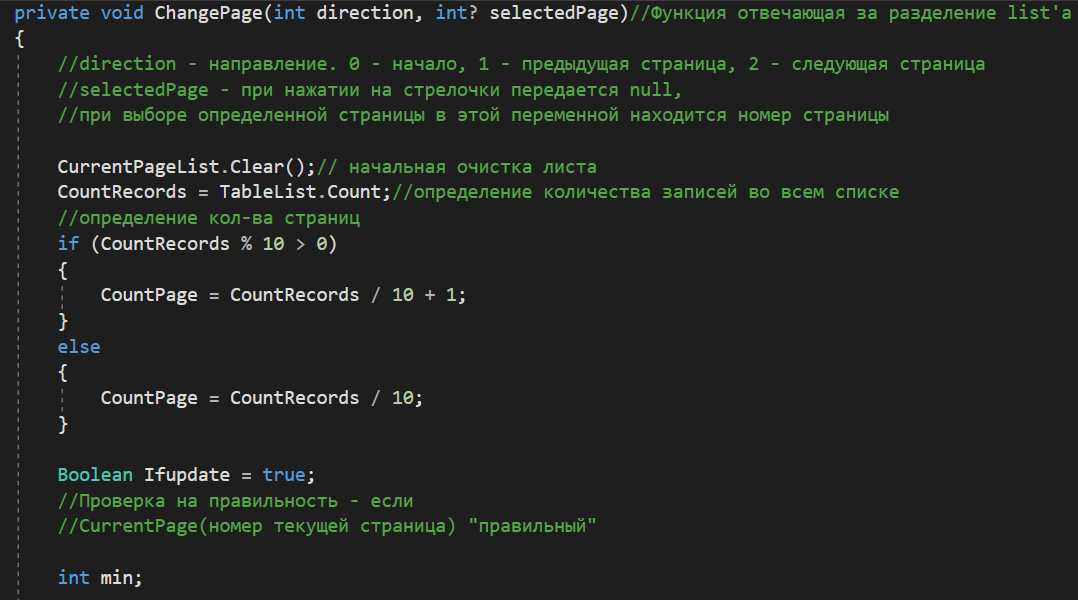
**x = 25;**

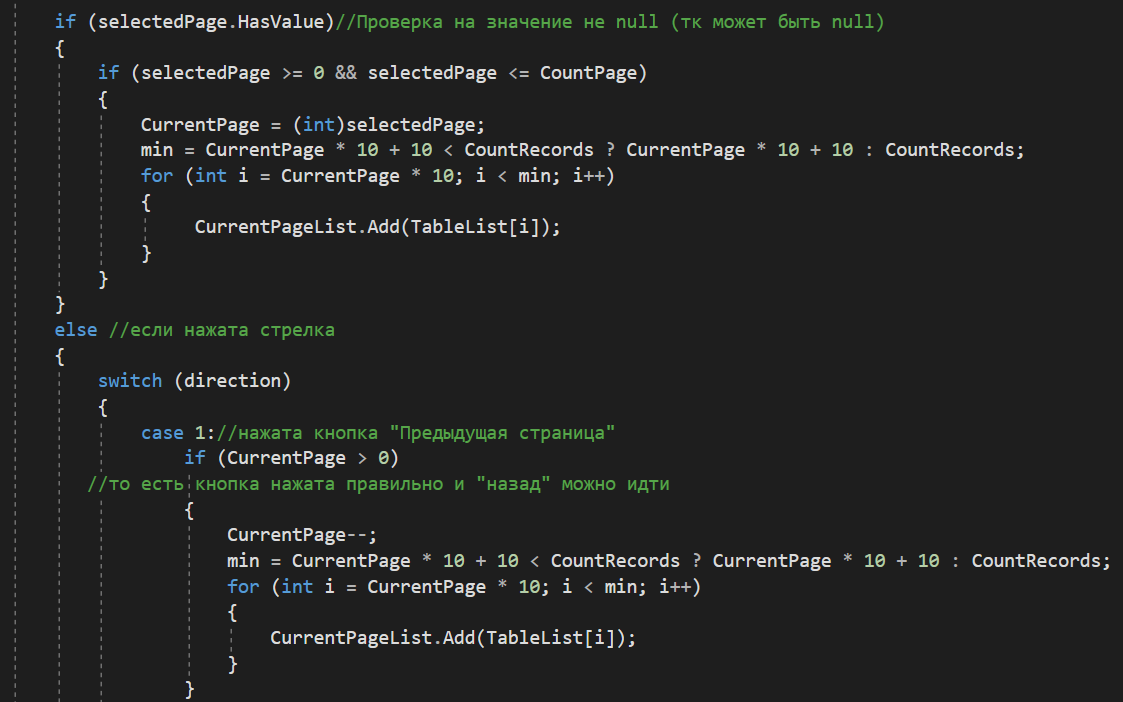
Свойство HasValue возвращает true, если объект хранит некоторое значение, и false, если объект равен null. Можем использовать это свойство для проверки наличия и получения значения.

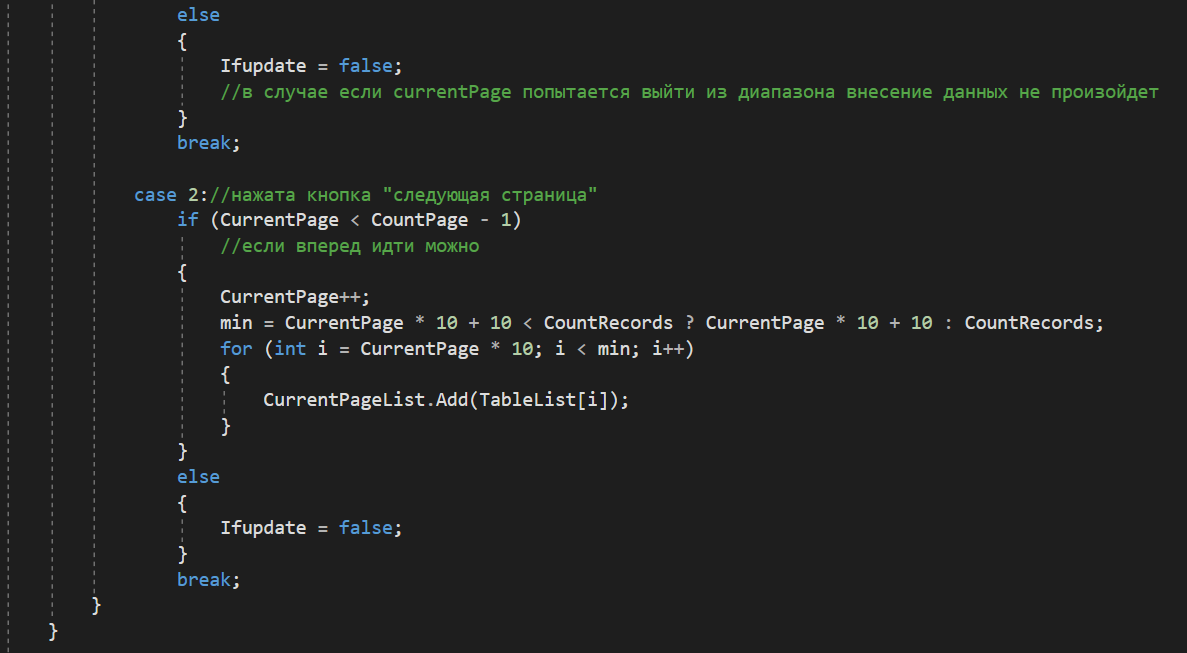
<https://metanit.com/sharp/tutorial/2.17.php>

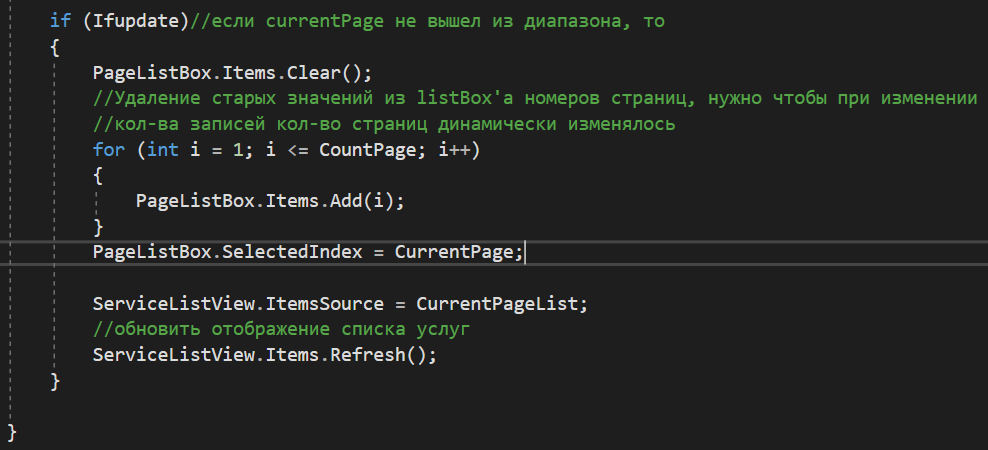
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Параметр selectedPage - при нажатии на стрелки передается null, при выборе определенной страницы в этой переменной находится номер страницы.

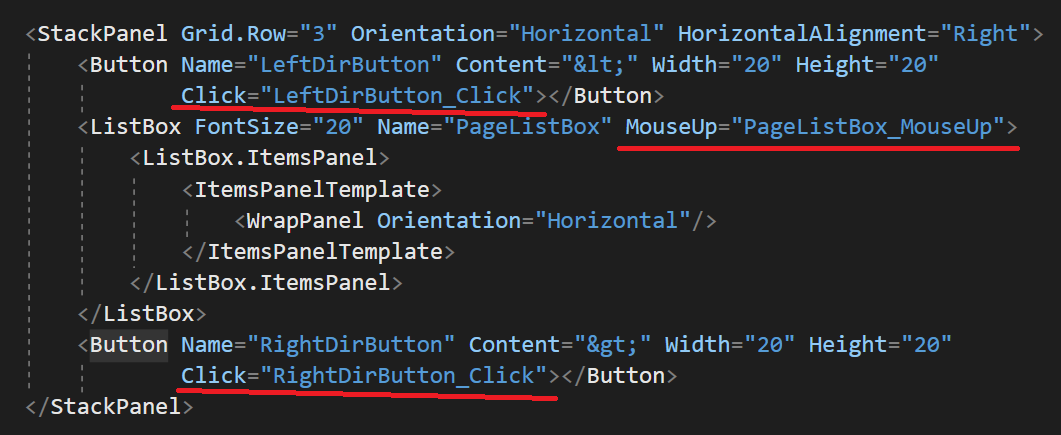




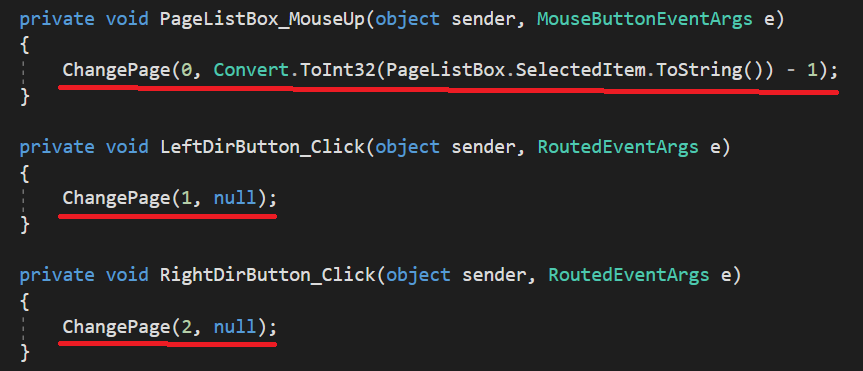




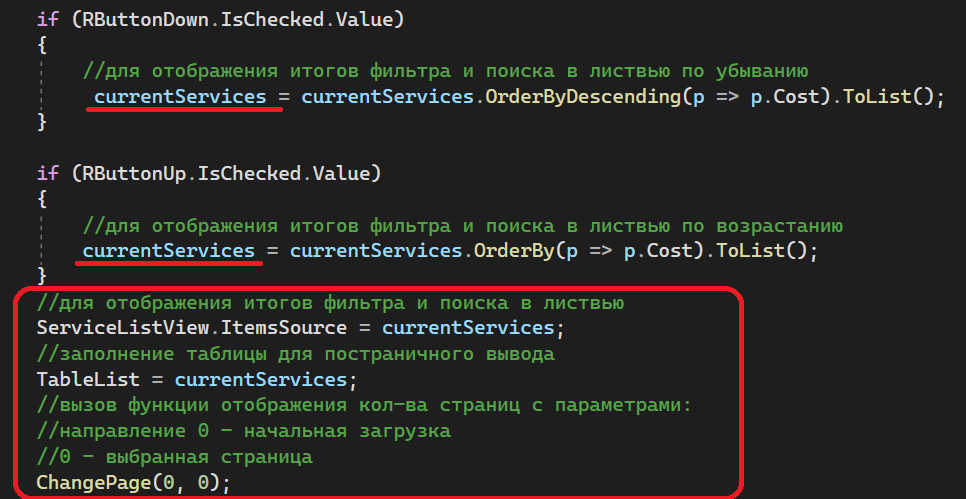
4.5. Чтобы кнопки заработали, требуется прописать в разметке методы-обработчики:



4.6. Теперь пропишем сами методы:

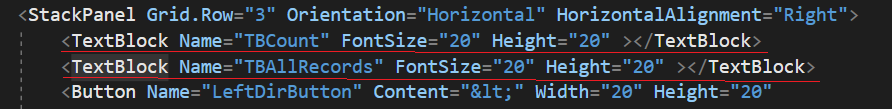


4.7. Для того чтобы корректно работало отображение страниц, требуется передавать в глобальную переменную TableList значения из основного листвью также и в функции UpdateServices() и совместить сортировку, фильтрацию и поиск с постраничным выводом. Поправим код этой функции:

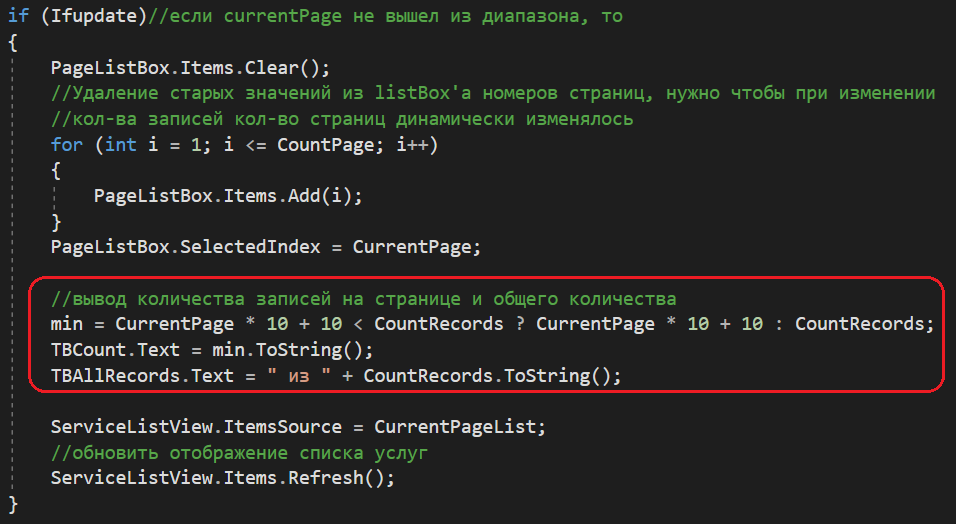


Запустить, проверить работу.

4.8. Осталось вывести количество записей на странице и общее количество записей. Для этого используем два TextBlock’а, поместив их в уже имеющийся StackPanel:



И пропишем в функцию ChangePage фрагмент кода для вывода информации:



Результат выполнения лаб.работы залить на гитхаб под новым коммитом

(7 лаба).

