Разработка формы «Запись клиента на услугу»

**1. Клонировать с гитхаба предыдущую версию проекта (итог лабы7)**

**2. Начнем с изменения таблицы Service нашей БД.**

*2.1. Откройте* файл service\_a\_import.txt, который Вы модифицировали в первой лабе *(если вдруг не сохранилось, измените тот файл заново из имеющегося исходника).*

*Далее описаны только! новые изменения файла (прошлые изменения этого файла у вас должны быть, если их нет, то повторите заново изменения из 1 лабы)*

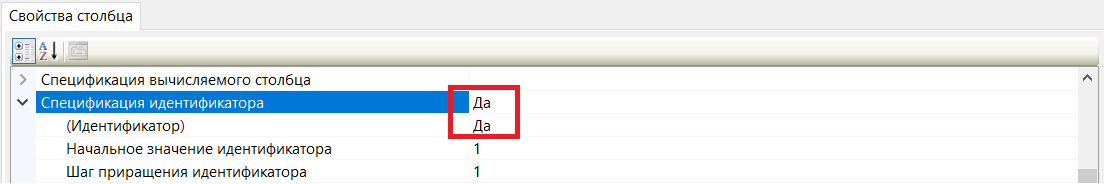
При помощи автозамены *замените* в поле Duration длительности в часах на длительности в минутах (не указывая единицу измерения) следующим образом:

|  |  |
| --- | --- |
| **Что заменить** | **На что заменить** |
| 1 час. | 60 |
| 2 час. | 120 |
| 3 час. | 180 |
| 4 час. | 240 |
| 5 час. | 300 |
| 6 час. | 360 |
| 7 час. | 420 |
| 8 час. | 480 |
| 9 час. | 540 |
| 10 час. | 600 |

Далее *заменить* автозаменой «мин.» на пусто.

*2.2 Выполнить* импорт неструктурированного файла service\_a\_import.txt в БД.

2.3. В полученной таблице переименовать поля, ПОЛЕ Duration сделать целого типа!!! *Добавить* поле ID, сделав его ключевым и указав для него автоинкремент:



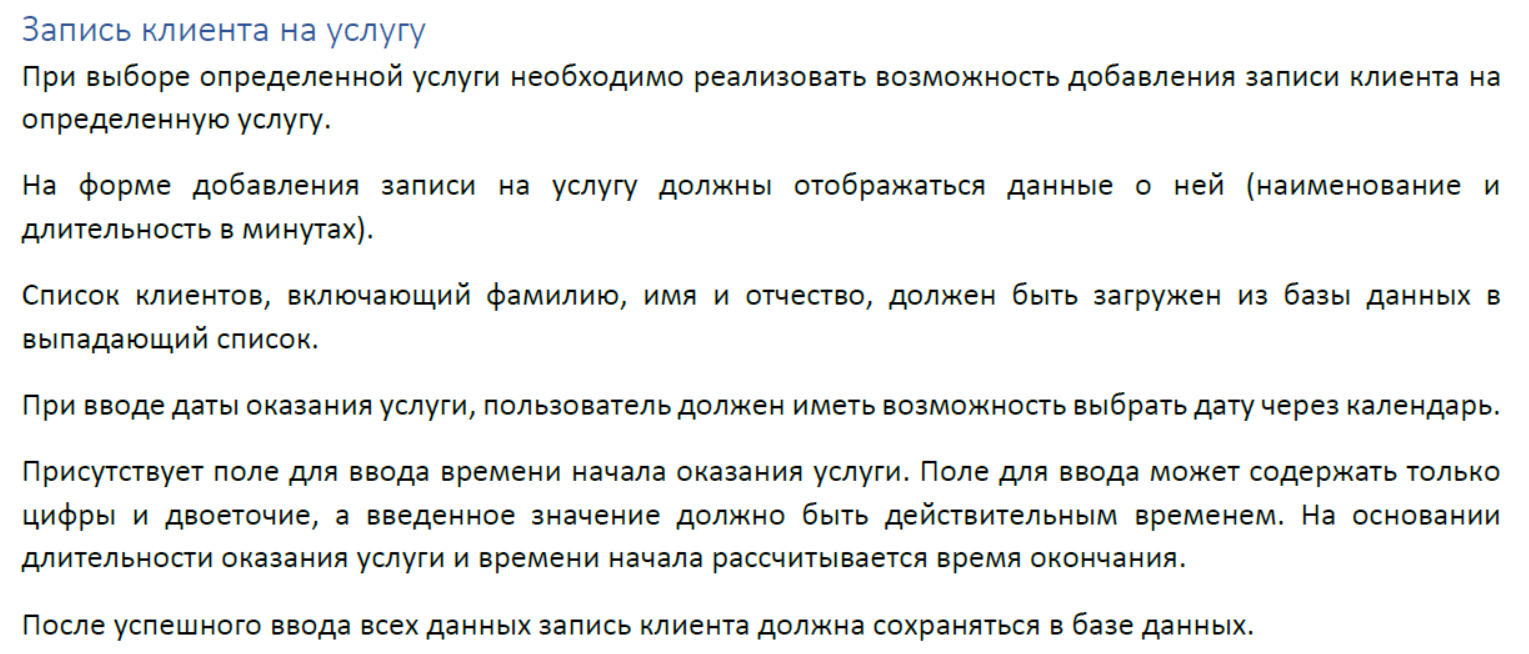
2.4. Через диаграмму в БД *удалить* связи таблицы Service, *сохранить* диаграмму, а затем *удалить* саму таблицу Service из БД.

*2.5. таблицу* Service\_a\_import *переименовать в* Service*. Восстановить* связи таблицы Service через диаграмму.

*Сохранить* БД.

*3. Из проекта нужно будет удалить БД совсем и подключить ее заново, как делали в лаб. работе 4.*

**4. Посмотрим Задание из файла Сессия1.pdf:**

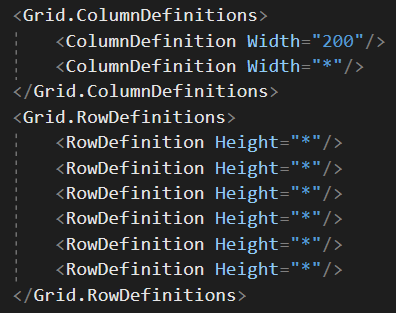


5. Добавим в проект новую страницу **SignUpPage** для реализации записи клиента на услугу.

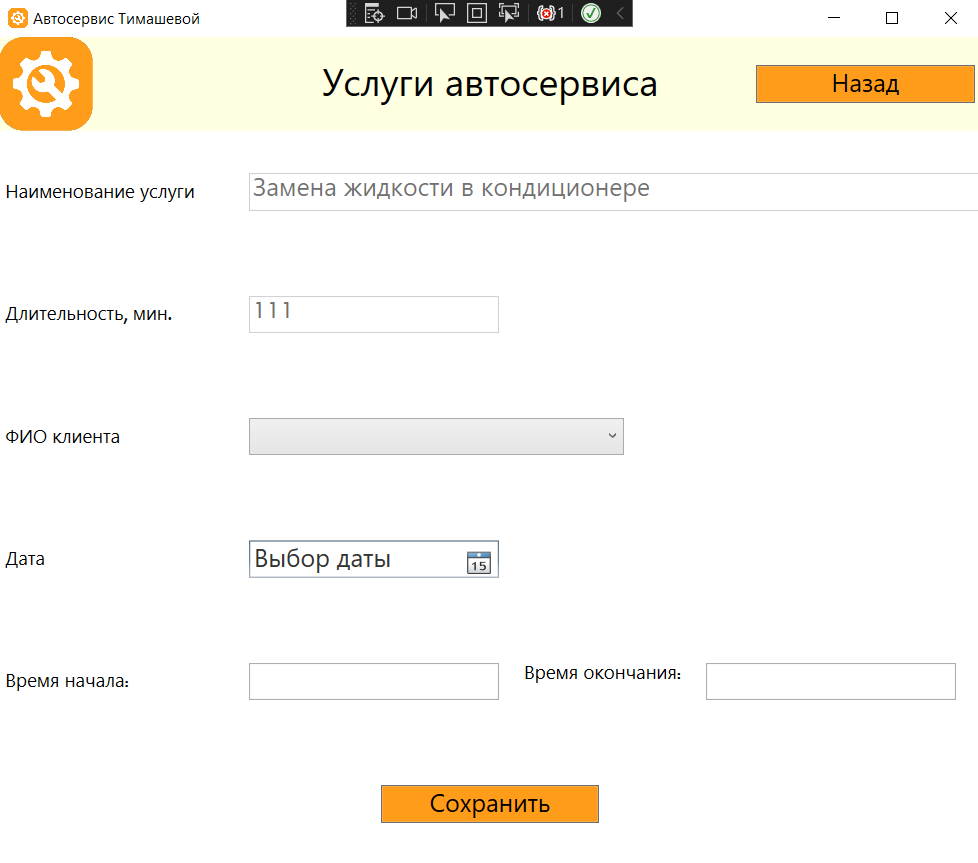
Назначьте ей шрифт:



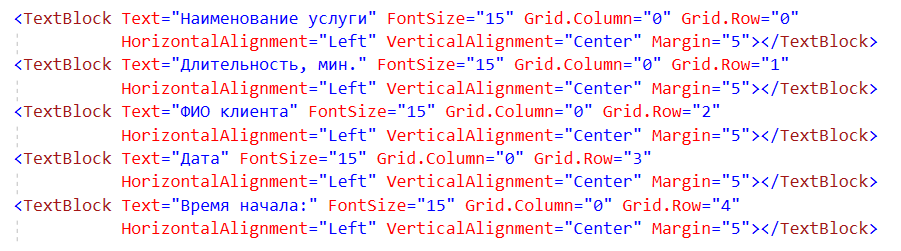
6. Сверстаем на ней сетку, состоящую из 6 строк и двух столбцов:



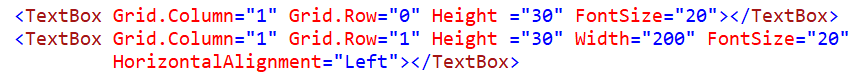
7. Выполним разметку в соответствии с рисунком:



Для этого разместим затем Текстовые блоки для отображения подсказок:



8. Опишем также TextBox для вывода наименования услуги и ее длительности:



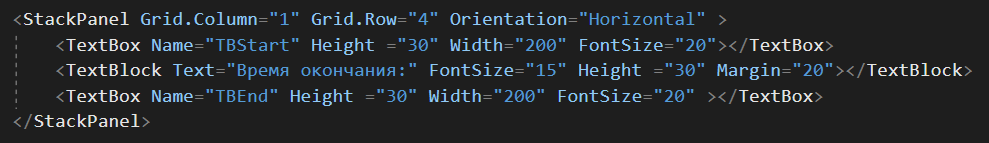
9. Опишем ComboBox для вывода ФИО клиента:



10. Опишем DatePicker для выбора даты оказания услуги:



11. Опишем элементы, объединив их в StackPanel: 2 текстовых поля для ввода времени начала/вывода времени окончания услуги; 1 текстовый блок с сообщением-подсказкой:

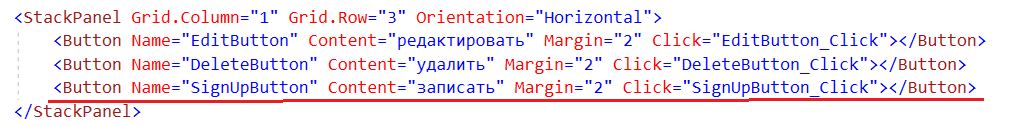


12. Добавим кнопку для сохранения изменений, дадим ей имя и укажем обработчик нажатия:

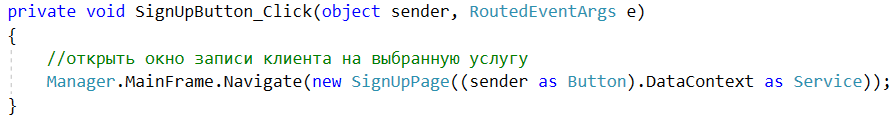


13. Cделаем переход на эту страницу со страницы ServicePage.xaml.

Для этого добавим кнопку «Записать» рядом с кнопками «Редактировать» и «Удалить» на странице списка услуг ServicePage.xaml:

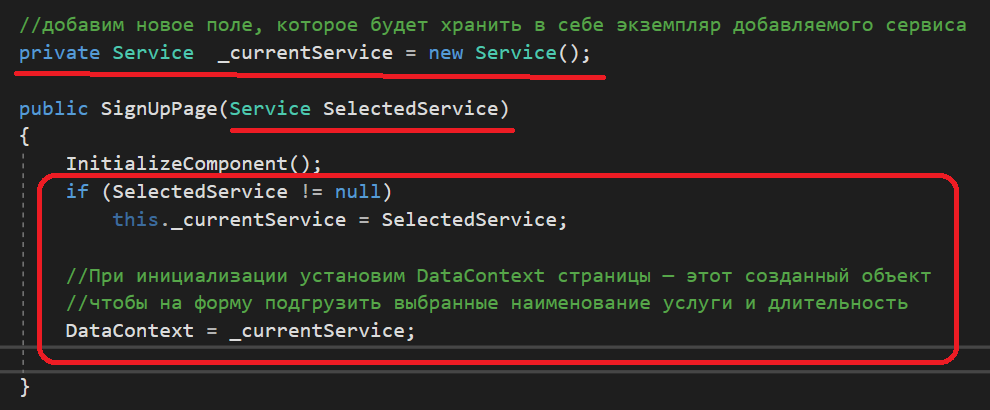


И пропишем обработчик события (это уже в коде ServicePage.xaml.cs)



**14. Перейдем к коду страницы SignUpPage.** На странице кода SignUpPage.xaml.cs добавим новое поле, которое будет хранить в себе экземпляр добавляемого сервиса (услуги). И при инициализации установим DataContext страницы — этот созданный объект.

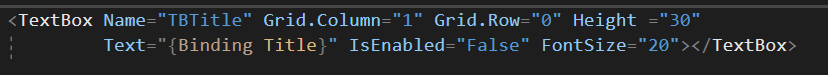
Также как и делали ранее для редактирования услуги, добавим параметр странице SignUpPage.xaml.cs. В него будем передавать экземпляр выбранной услуги и, в случае если он не пустой, присваивать нашему полю CurrentService.



Запустить, проверить работу перехода на форму и разметку формы.

15. Затем, используя привязку данных, укажем, какому свойству обращаться к каждому элементу при загрузке данных на странице SignUp.

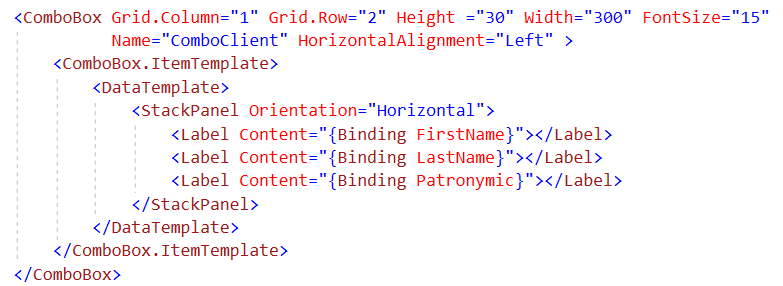
Например, свойство Text у первого TextBox'а будет обращаться к названию услуги, используя привязку (binding)*,* а также установим для поля невозможность внесения изменений в нем через свойство IsEnabled:



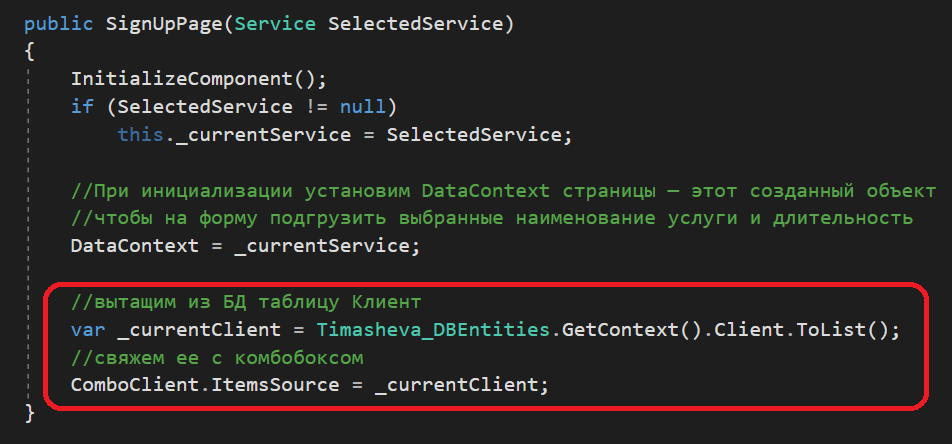
Аналогично пропишите привязку для следующего TextBox'а к полю Длительность, и не забудьте также установить для него невозможность внесения изменений.

16. Для того чтобы в ComboBox отображались ФИО клиента (три поля таблицы Client), нужно сделать следующее:

Поправим разметку: зададим ComboBox имя, также укажем отображаемые в нем элементы, т.к. их много (фамилия, имя, отчество клиента) воспользуемся свойством DataTemplate:

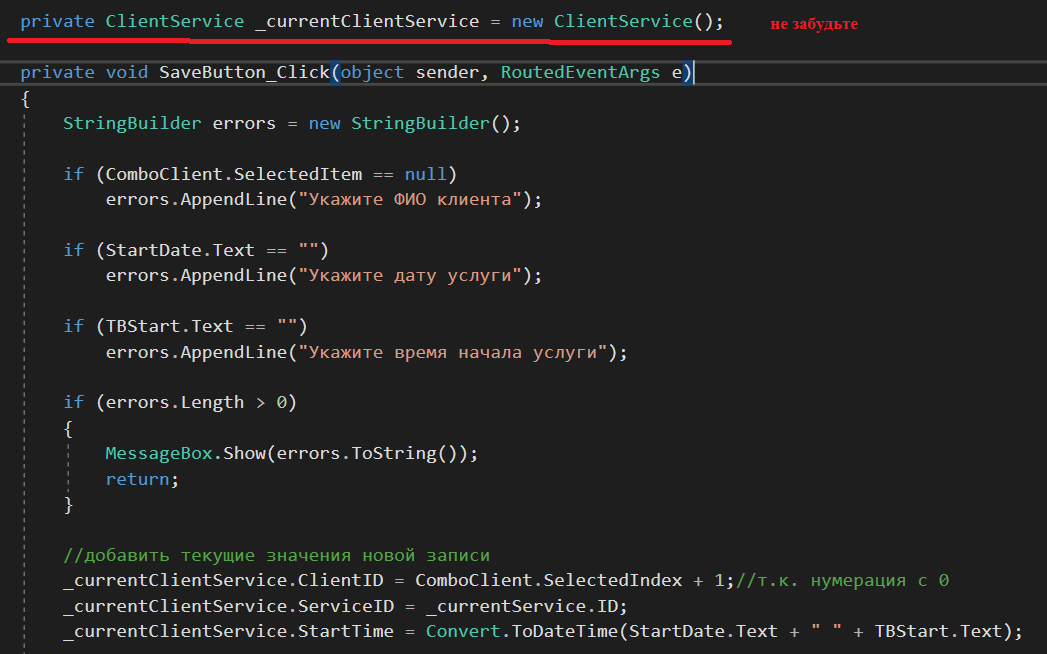


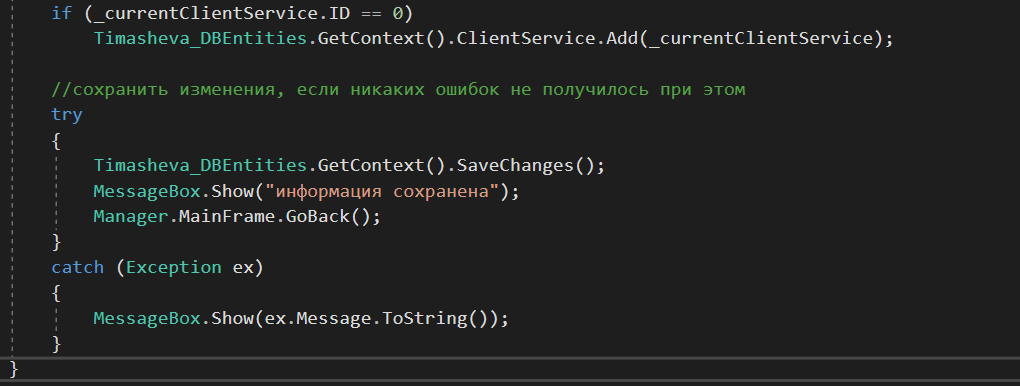
На странице кода SignUpPage.xaml.cs добавим при инициализации параметр, который будет хранить в себе экземпляр сущности Клиент и установим его в качестве источника для строк комбобокса:



17. Далее обработаем нажатие на кнопку «Сохранить» и в коде пропишем логику обращения к модели данных и добавления нового экземпляра услуги:

Код обработчика:

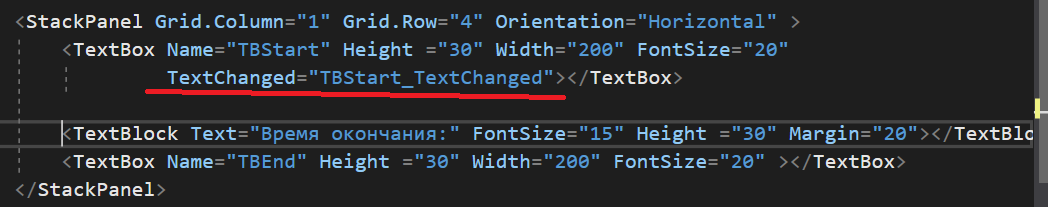




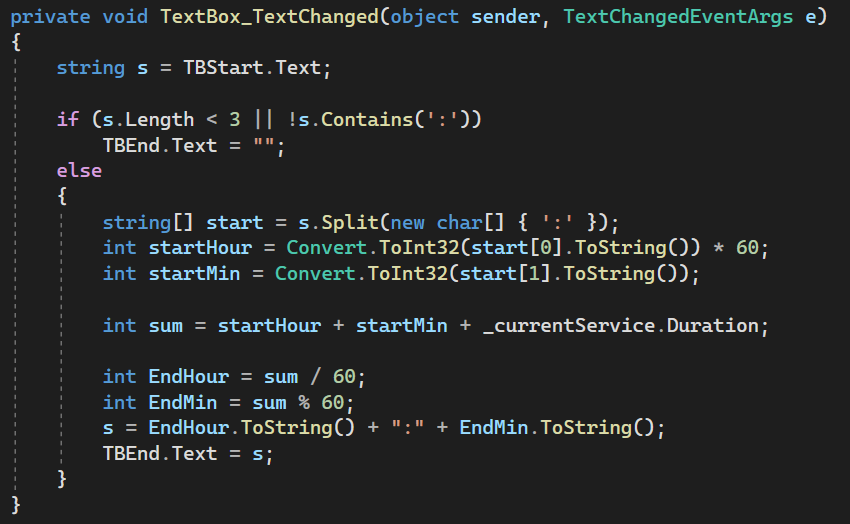
Запустить, проверить работу.

18. Надо сделать, чтобы на форме шел автоматический расчет времени окончания услуги.

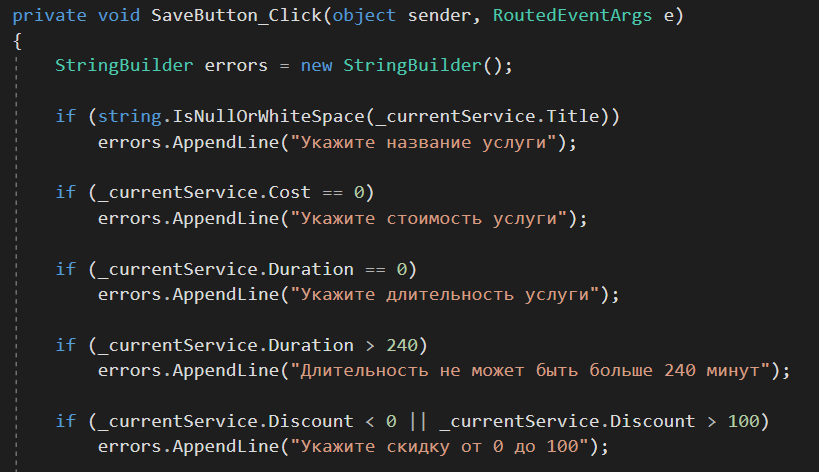
Добавим свойство:



В коде пропишем обработчик события по вычислению времени окончания услуги:

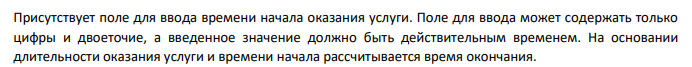
****

19. ***Поправим*** некоторые строчки кода в AddEditPage.xaml.cs (добавим защиту на то, что длительность не может быть более 240 минут и менее 0 самостоятельно):



Запустить, проверить работу.

**20. Реализовать невозможность ввода некорректных символов в поле Время начала должно не вводиться ничего кроме цифр и двоеточия**

****

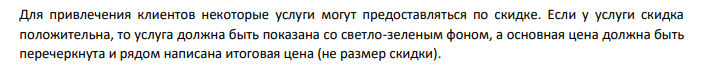
**21. Реализовать:**

****

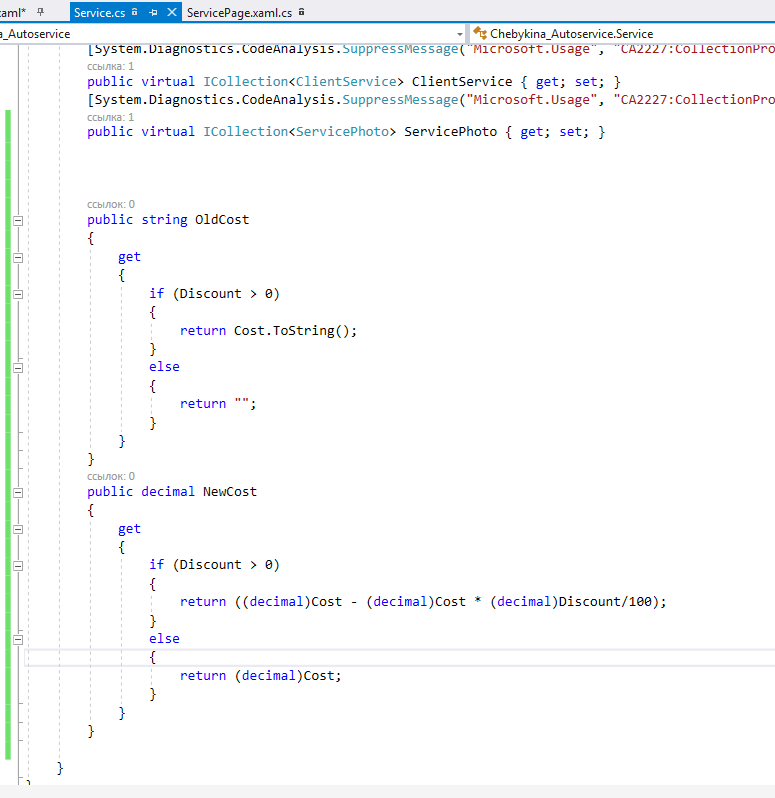
**подсказка:**

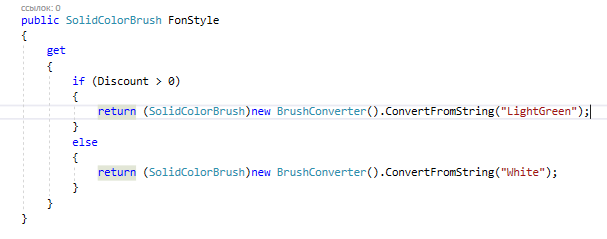
****

**22. Реализовать:**

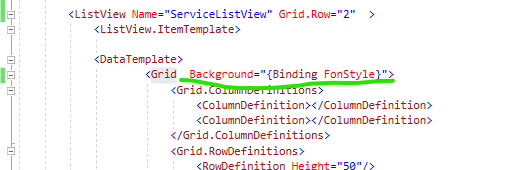
****

Service.cs – класс таблицы Service, добавить туда поля:





Разметка ServicePage:



и



<https://www.nookery.ru/how-to-change-the-background-color-background-c-wpf/>

Результат выполнения лаб.работы залить на гитхаб под новым коммитом (8лаба).