

Практическая работа №6

Реализация уведомления об изменении свойств объектов

1 Цель работы

- 1.1 Научиться уведомлять связанные с объектами элементы управления о изменении значений свойств объектов;
- 1.2 Научиться реализовывать интерфейс INotifyPropertyChanged.

2 Литература

- 2.1 <https://docs.microsoft.com/en-us/dotnet/desktop/winforms/how-to-implement-the-inotifypropertychanged-interface?view=netframeworkdesktop-4.8>
- 2.2 <https://metanit.com/sharp/wpf/11.2.php>

3 Задание

3.1 Реализация интерфейса INotifyPropertyChanged

3.1.1 Создать оконное приложение и добавить в него класс Пользователь с закрытыми полями логин и пароль.

Используя быстрые действия и рефакторинг:

- инкапсулировать поля (ПКМ на название поля)
- использовать тело блока для методов доступа (ПКМ на set)

3.1.2 Указать, что класс Пользователь реализует интерфейс INotifyPropertyChanged, и реализовать его, используя быстрые действия и рефакторинг.

3.1.3 Создать метод, запускающий событие PropertyChanged:

```
protected void NotifyPropertyChanged([CallerMemberName] string propertyName = null)
{
    PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(propertyName));
}
```

3.1.4 В свойстве Логин и Пароль после изменения значения поля вызвать метод NotifyPropertyChanged:

```
NotifyPropertyChanged();
```

3.2 Редактирование данных в отдельной форме

3.2.1 Добавить в форму поле users типа List<T> (вместо T указать созданный класс) и заполнить его данными о пользователях при загрузке формы.

3.2.2 Добавить на форму DataGridView и при загрузке формы привязать его к списку пользователей:

```
контрол.DataSource = список;
```

3.2.3 Добавить в приложение форму редактирования, принимающую в параметрах объект типа Пользователь.

Реализовать открытие формы редактирования при событии CellDoubleClick в DataGridView. Для получения объекта списка по его номеру можно использовать следующую форму:

```
список.ElementAt(номер элемента в списке)
```

3.2.4 Добавить на форму редактирования два поля ввода для отображения логина и пароля и поле current для хранения текущего пользователя.

3.2.5 В конструкторе формы редактирования:

- присвоить полю current значение переданного через конструктор пользователя;
- присвоить полям ввода логин и пароль текущего пользователя.

3.2.6 На событие FormClosing формы редактирования присвоить логину и паролю текущего пользователя данные из полей ввода, если они отличаются от исходных.

3.3 Обработка события PropertyChanged для выделения цветом измененной строки

3.3.1 Добавить в главную форму поле для хранения номера выбранной строки. Значения поля менять в событии CellDoubleClick.

3.3.2 Добавить на главной форме элементу списка, полученному при CellDoubleClick, подписку на событие PropertyChanged.

3.3.3 При наступлении события перебрать все строки DataGridView и изменить на белый цвет фона все строк.

строка.DefaultCellStyle.BackColor = Color.Цвет;

3.3.4 После смены цвета фона на белый поменять на #FFCC66 фон строки DataGridView с номером из п.3.3.1.

Для получения цвета из HEX используется следующий синтаксис:

ColorTranslator.FromHtml("#FFCC66")

3.4 Применение BindingList<T>

3.4.1 Изменить тип поля users на BindingList<T> (вместо T указать созданный класс).

BindingList<T> – список, поддерживающий привязку данных. При изменении свойств сообщает об этом элементу управления, к которому привязан

3.4.2 Добавить полю users подписку на событие ListChanged перед заполнением списка при загрузке главной формы.

3.4.3 При наступлении события ListChanged перебрать все строки DataGridView и, если у строки есть связанный объект, изменить цвет текста строк с некорректным паролем (до 6 символов) на цвет в формате RGB (200; 10; 0). У корректных паролей цвет текста – черный.

Получение свойства объекта на основе строки DataGridView:

((item as DataGridViewRow).DataBoundItem as ТипДанных)?.Свойство

Для получения цвета из RGB используется следующий синтаксис:

Color.FromArgb(200; 10; 0)

4 Порядок выполнения работы

4.1 Выполнить все задания из п.3 в одном решении PractWork6. Возможные ошибки требуется обрабатывать. Выполнить форматирование и рефакторинг кода.

4.2 Ответить на контрольные вопросы.

5 Содержание отчета

5.1 Титульный лист

5.2 Цель работы

5.3 Ответы на контрольные вопросы

5.4 Вывод

6 Контрольные вопросы

6.1 Какой интерфейс требуется реализовать, чтобы объект мог оповещать об изменении значений своих свойств?

6.2 Какое пространство имен требуется подключить для работы INotifyPropertyChanged?

6.3 Для чего используется атрибут [CallerMemberName]?

6.4 Для чего используется класс `BindingList<T>`?