**Лабораторная работа 9. Преобразование и подтверждение данных**

**ЗАДАНИЕ**

**#9A**

Реализуйте интерфейс IMultiValueConverter для создания конвертера, который принимает целые числа, указывающие день, месяц и год, и возвращает объект DateTime. Также реализуйте метод ConvertBack.

<Window.Resources>

<local:DateMultiValueConverter x:Key="DateConverter" />

</Window.Resources>

Реализуйте правила, которые подтверждают данные, используемые конвертером IMultiValueConverter. Введите такие правила, чтобы нельзя было задавать несуществующие даты (например, 31 июня). Кроме того, реализуйте правило, которое подтверждает 29 февраля только в високосном году.

**#9B**

**Создайте приложение, эмулирующее работу троллейбусного парка.**

При загрузке окна пользователь должен ввести количество троллейбусов и подтвердить выбор. «Троллейбусы» появятся на форме. Далее пользователь выбирает троллейбус и «запускает» его на линию, при этом определяя начало работы парка. Каждый троллейбус отстает от предыдущего на какое-то время. Если троллейбус уже выехал, он не может выехать еще раз. Все данные о движении транспорта записываются в лог.

1. Создайте класс TrolleyBus (троллейбус), который содержит в себе закрытое статическое поле startWorkingTime, являющееся полем только для чтения, а также поле и свойство для номера троллейбуса (типа int).

В классе должны быть определены:

* статический конструктор по умолчанию, присваивающий статическому полю startWorkingTime текущую дату и время;
* конструктор по умолчанию, задающий фиксированное количество троллейбусов (позже это значение всегда будет привязано к textbox).
* конструктор с параметром, задающий номер троллейбуса;
* метод Drive, возвращающий строку с информацией какой троллейбус выехал, время выезда и его отставание от начала работы парка.

1. Создайте приложение WPF (пример реализации на рис.1-3). Добавьте проверку на корректность вводимых данных (в TextBox) с помощью системного класса ValidationRule и переопределения метода ValidationResult (рис.4-6).

Добавьте в проект класс TrolleyBusRule, наследующий ValidationRule, в XAML укажите привязку к этому классу. Обязательно для проверки:

* Троллейбусов в парке от 2 до 60;
* Нельзя вводить буквы;
* Кнопка «Подтвердить» не активна, пока в textbox не введено корректное значение.

Пример реализации:

<https://docs.microsoft.com/en-us/dotnet/framework/wpf/data/how-to-implement-binding-validation>

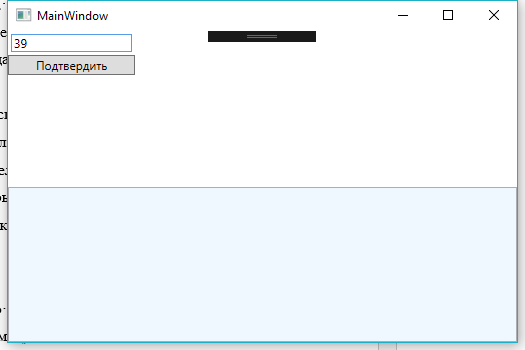


Рис.1. Первый запуск страницы

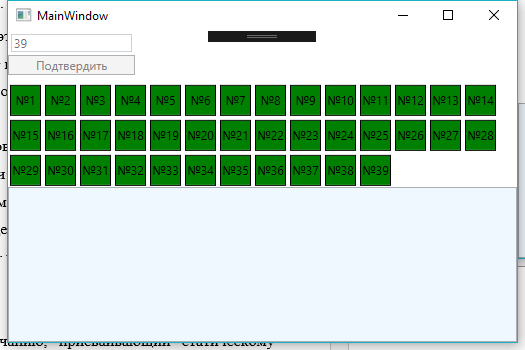


Рис. 2. Формирование «троллейбусного парка»

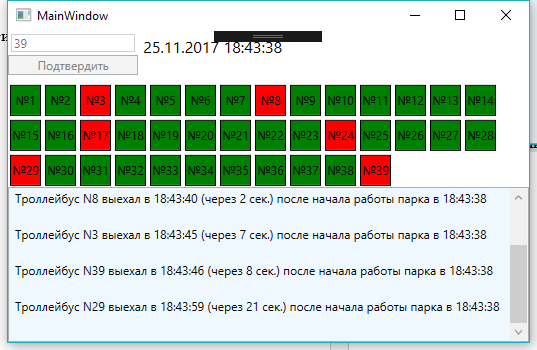


Рис. 3. Работа приложения

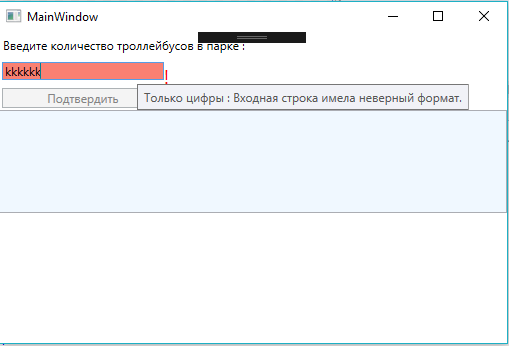


Рис. 4. Ошибка ввода

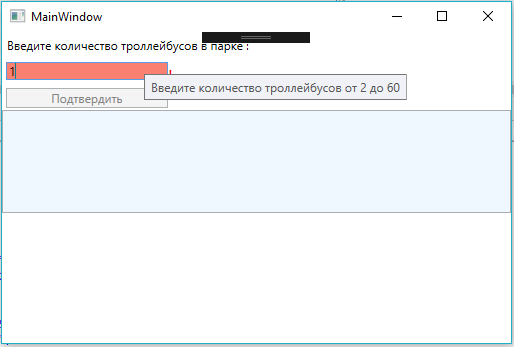


Рис. 5. Ошибка ввода

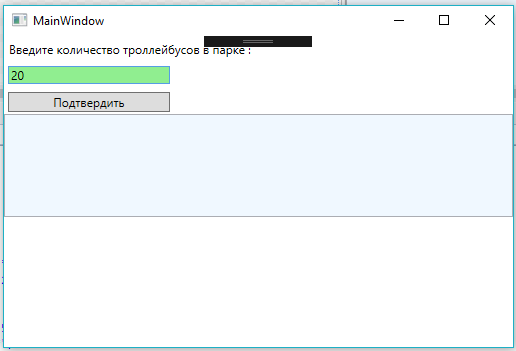


Рис. 6. Верные данные

**#9С**

С помощью класса ObservableCollection создайте две настраиваемые коллекции данных объектов, Служащий и Отдел, где объекты Отделы располагают связью «один-ко-многим» с объектами Служащие. Реализуйте в каждом объекте интерфейс INotifyPropertyChanged. Создайте пользовательский интерфейс, отображающий объекты Служащий, сгруппированные по объектам Отдел, где они расположены.