**Лабораторная работа по курсу**

**«WPF»**

**Тема: Элементы управления**

**Цель:**

Закрепить у слушателей практические навыки и теоретические знания при работе с WPF.

**Необходимые инструменты:** MS Visual Studio 2013

**Документация:** Конспект, Литература

**Задание 1.** **Приложение «Секундомер»**

Спроектировать интерфейс и разработать программу «Секундомер». Вариант (функциональных возможностей) представлен на рисунке. Интерфейс (компоновку) и оформление придумать самостоятельно. Функциональность можно расширять.

Особенности:

- при изменении размеров приложения элементы управления должны адекватно реагировать;

- при остановке секундомера должна активизироваться кнопка «Сброс», приводящая приложение в исходное положение;

- при сворачивании приложения программа должна размещаться в трее.

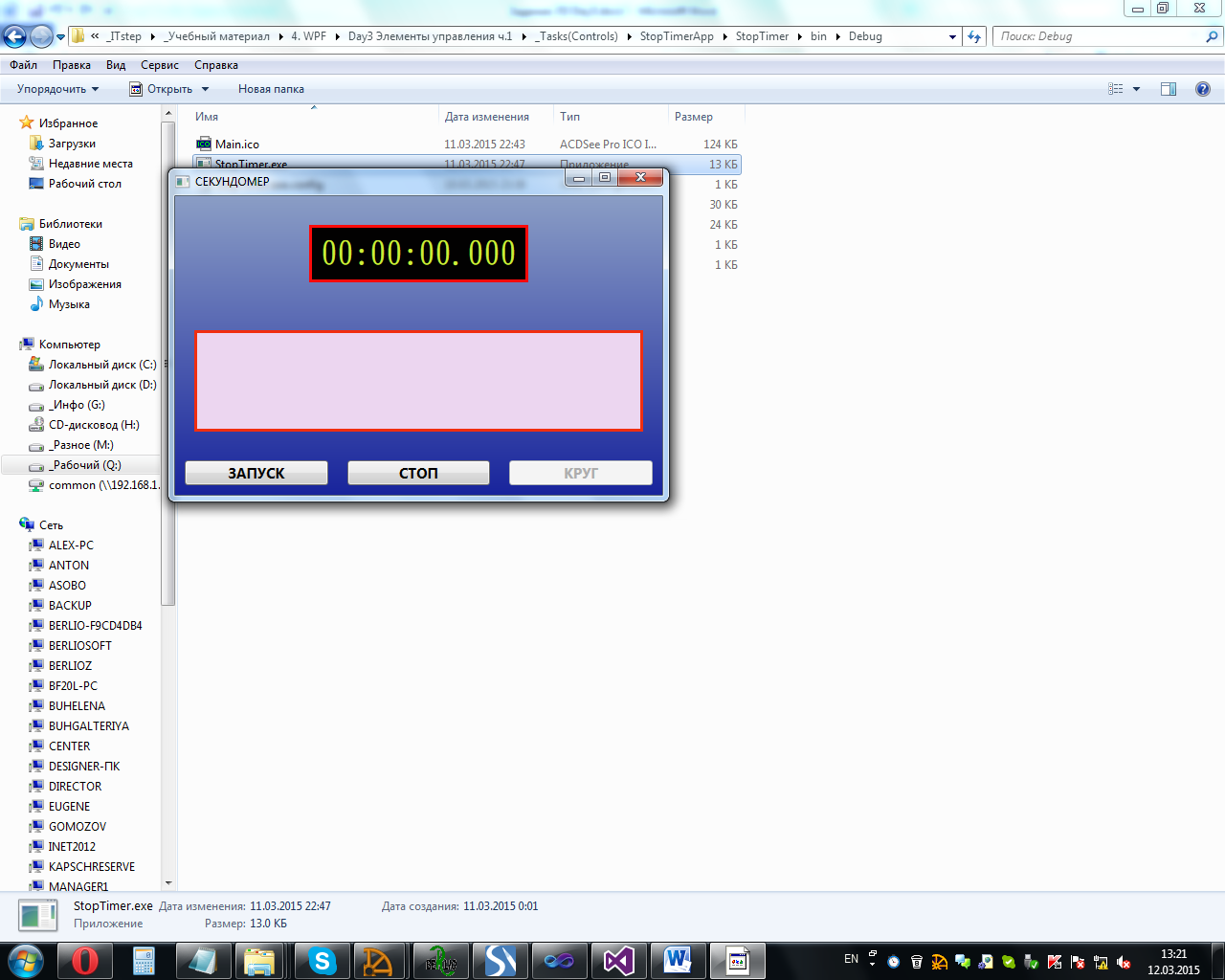
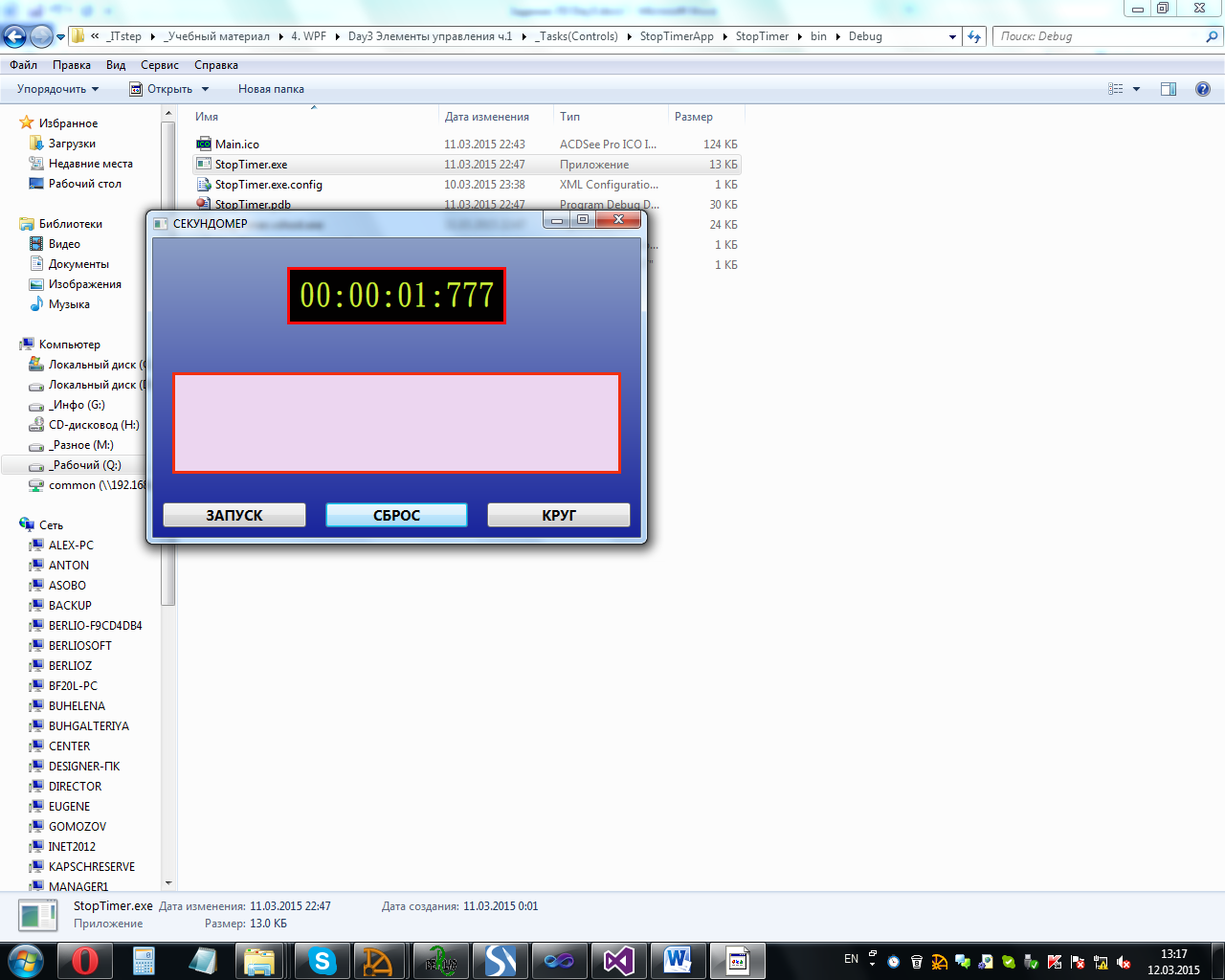
 

Рисунок Исходное положение Рисунок В режиме работы

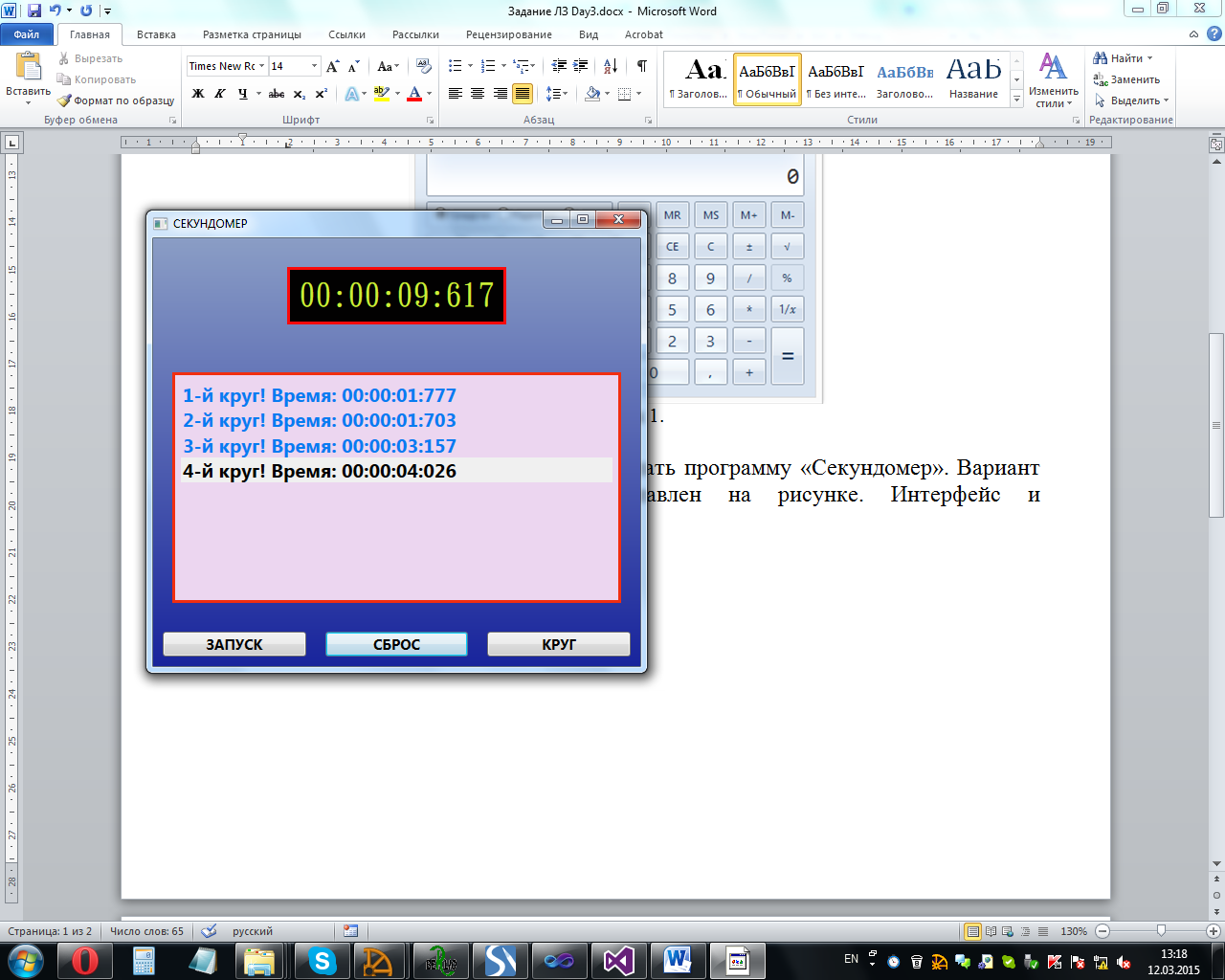


Рисунок Добавление кругов.

**Дополнительно:**

- Для реализации таймера в программе необходимо использовать класс DispatcherTimer (по функциональности данный класс аналогичен классу Timer из библиотеки Windows Forms);

- для реализации работы программы необходимо использовать класс TimeSpan. (данный класс позволяет установить интервал таймера, а также определить количество часов, минут, секунд между двумя датами типа DateTime)

- для вывода значений минут, секунд, часов в формате 00:00:00 необходимо использовать формат вывода:

***String.Format("{0:00}:{1:00}:{2:00}:{3:000}", Hours, Minutes, Seconds, Milliseconds);***

- для реализации сворачивания в трей необходимо использовать класс System.Windows.Forms.NotifyIcon. (пример, размещения приложения WPF в трее представлен по ссылке)

<https://ebbypeter.wordpress.com/2010/06/28/minimize-a-wpf-application-to-system-tray-in-c/>

**Задание 2.** Приложение «Простейший проигрыватель»

Разработать приложение «Простейший проигрыватель», позволяющий просматривать видео файлов. Проигрыватель должен поддерживать следующие режимы: «Play», «Pause», «Stop». Открытие видеофайла реализовать с использованием стандартного диалога с настройкой соответствующих фильтров. В плейере должна быть возможность прокрутить фильм и регулировать громкость звука.

**Дополнительно:**

1. Реализовать интерфейс адекватно реагирующий на изменение размеров формы.
2. Реализовать плей-лист воспроизводимых файлов.
3. Реализовать автоматический переход на следующий видеофайл.
4. Настройки параметров отображения.

**Приложение:**

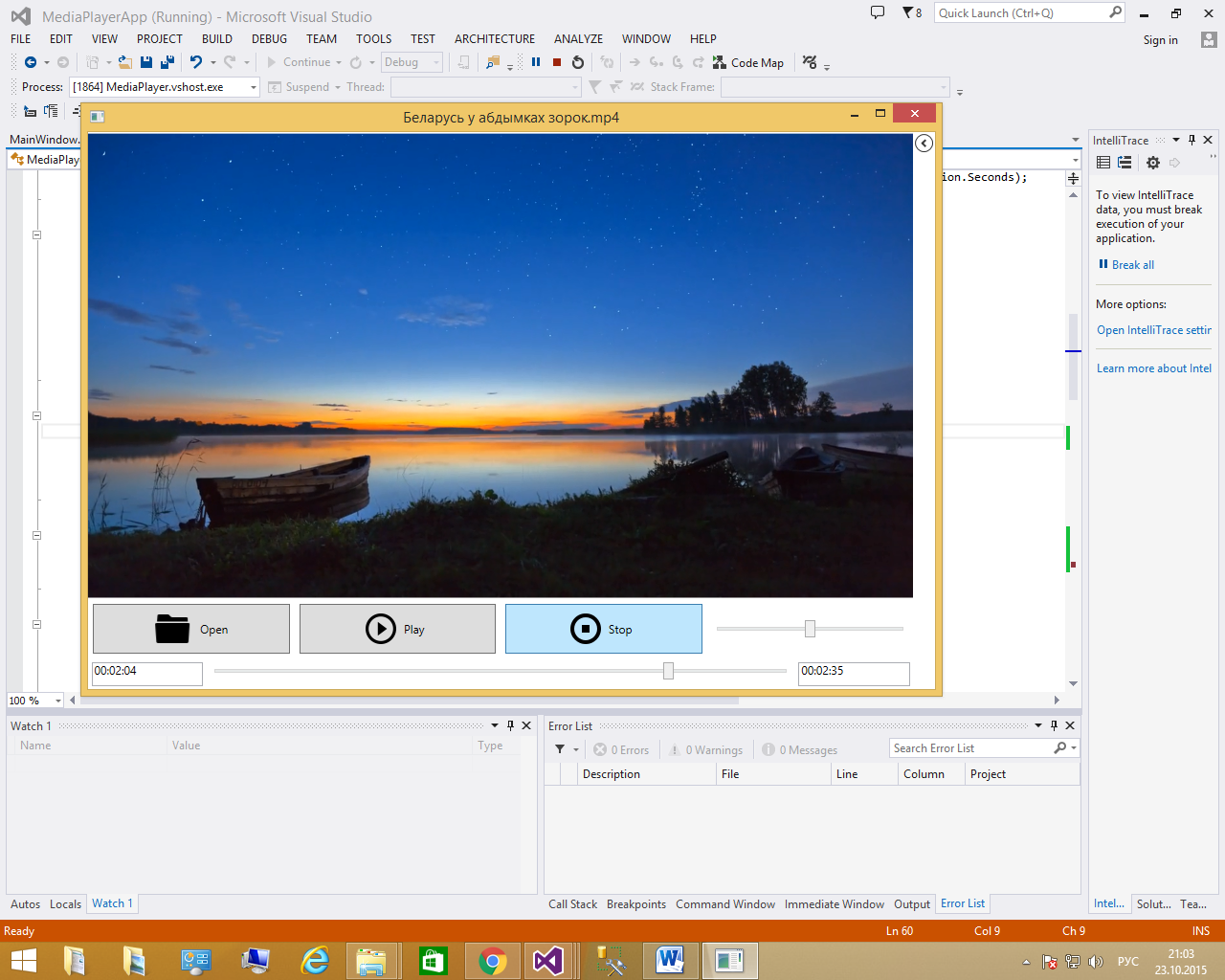
1. Для создания проигрывателя используем MediaElement
2. Для указания воспроизводимого файла используем свойство Source.
3. Для запуска, остановки и паузы использовать соответствующие методы: Play(), Stop(), Pause().
4. Важно! Для использования методов Stop(), Pause() – устанавливаем свойство элемента MediaElement в значение Manual.

Свойство LoadedBehavior – устанавливает состояния для мультимедийного файла. Данное свойство содержит значения: Stop, Close, Play, Manual, Pause.

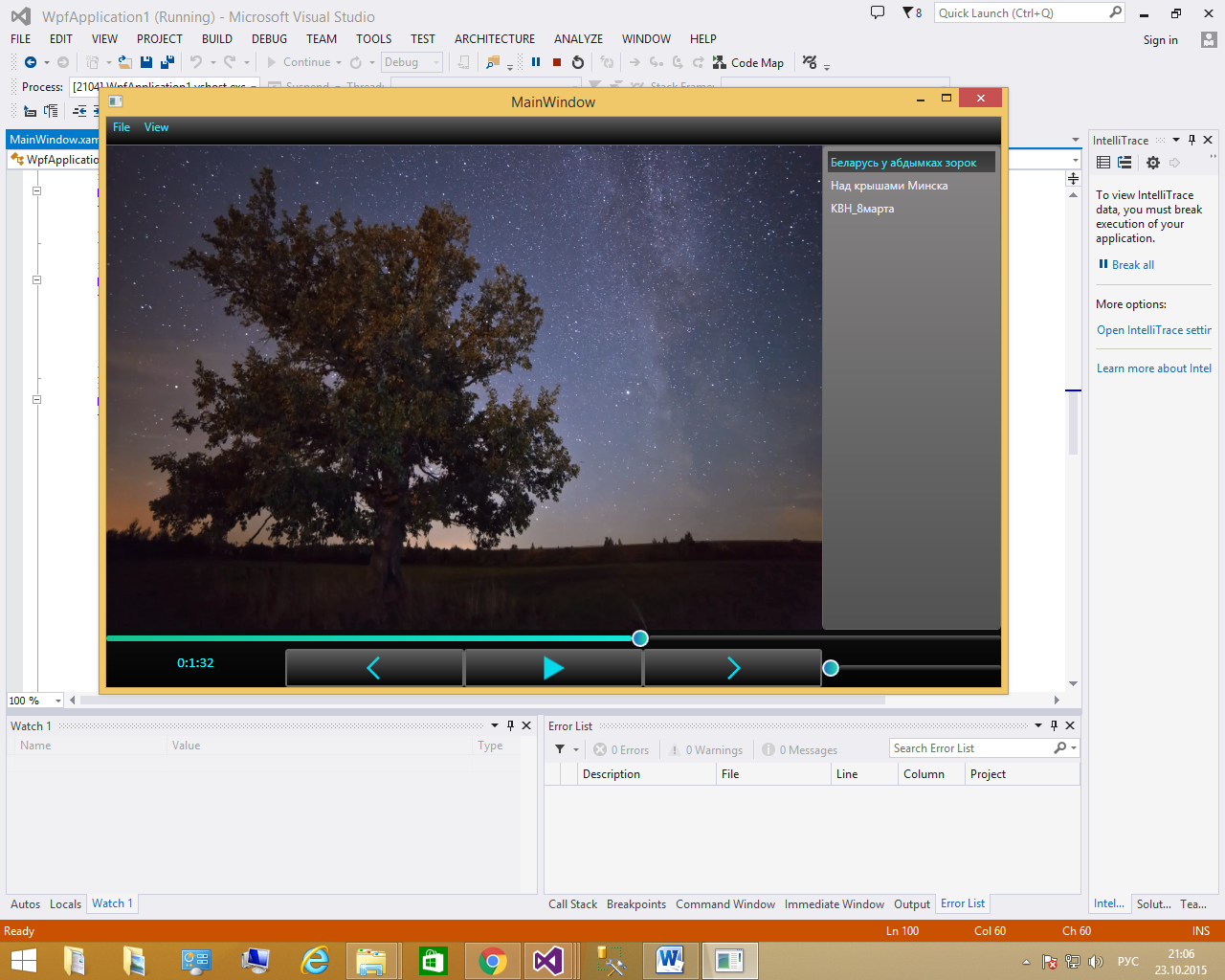
1. Свойство NaturalDuration – продолжительность мультимедийного файла.
2. Для организации работы MediaElement использовать свойства: MediaOpened, MediaEnded, MediaFailed.
3. Дополнительные свойства: HasAudio, HasVideo, IsMuted, NaturalVideoHeight, NaturalVideoWidth, SpeedRatio, Volume.

* Для вывода времени использовать формат: String.Format("00:00:00")
* Для изменения позиции воспроизведения использовать элемент Slider и его обработчики события PreviewMouseLeftButtonUp и PreviewMouseLeftButtonDown. Событие PreviewMouseLeftButtonUp – это события возникает при нажатии левой кнопки мыши при условии, что указатель мыши находится над этим элементом.

Примерный вариант приложения (по функционалу):



Пример



Пример плей-листа в проигрывателе

Внимание! Графическое оформление будем изучать на следующем уроке