

Лабораторное занятие 34

Часы-будильник-календарь

1. Создайте новый проект **Будильник Тимашевой Эльзы**.

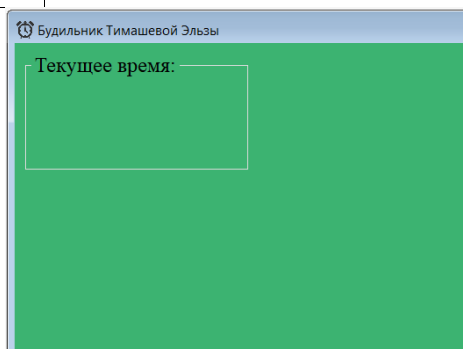
Произведем настройку формы:

Свойство	Значение
Text	Будильник Тимашевой Эльзы
BackColor	Ваш вариант
Icon	alarm.ico / ваш вариант
Font	Times New Roman, 14pt
MinimumSize	500;500
Size	600;500
StartPosition	CenterScreen
AutoScaleMode	Dpi
Name	AlarmForm

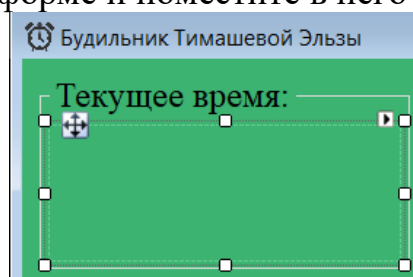
2. Поместите на форму элемент **GroupBox**.

Задайте следующие значения его свойств:

Size	200;100
Text	Текущее время:
Name	TimeBox



3. Выделите **TimeBox** на форме и поместите в него компонент **Panel**.

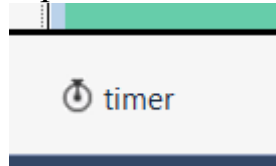


Задайте следующие значения свойств панели:

BorderStyle	FixedSingle
Size	190;60
Шрифт	Times New Roman, полужирный, 28 пт
Name	TimePanel

4. Теперь можно приступать к выводу на панель текущего времени (в TimeBox). Для этого нам понадобятся элементы из классов Timer и Label.

Поместим на форму таймер, заменив его свойство **Name = timer**:

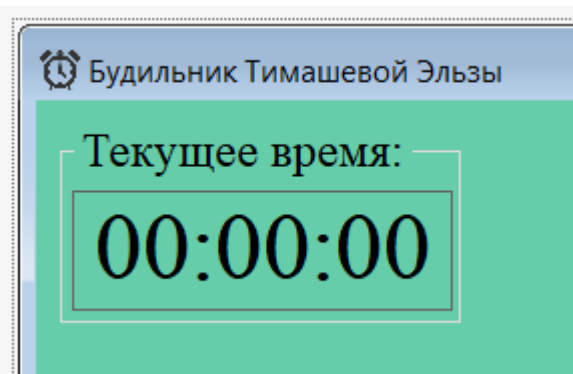


Элемент Label помещаем в TimePanel. Label будет служить для вывода времени на панель текущего времени TimePanel в TimeBox.

Установим свойства для **Label**:

Name как **timeLabel**,

Text как 00:00:00



Таймер позволит нам генерировать событие OnTimer с частотой, определяемой его свойством Interval. По умолчанию оно равно 1000 миллисекунд (или одной секунде), что вполне нас устроит. Когда мы вызываем обработчик события OnTimer, происходит сброс таймера и отсчёт промежутка времени, определённого свойством Interval, начинается снова.

Чтобы получить текущее время, воспользуемся структурой DateTime. Она представляет собой текущее время, обычно выраженное как дата и время суток.

Свойства:

Имя	Описание
Date	Возвращает компоненту даты этого экземпляра.
Day	Возвращает день месяца, представленный этим экземпляром.
DayOfWeek	Возвращает день недели, представленный этим экземпляром.
DayOfYear	Возвращает день года, представленный этим экземпляром.
Hour	Возвращает компонент часа даты, представленной этим экземпляром.
Kind	Возвращает значение, указывающее, на каком времени основано время, представленное этим экземпляром: местном, UTC или ни на том, ни на другом.
Millisecond	Возвращает компонент миллисекунд для даты, представленной в данном экземпляре.
Minute	Возвращает компонент минуты даты, представленной этим экземпляром.

Month	Возвращает компонент месяца даты, представленной этим экземпляром.
Now	Возвращает объект DateTime, которому присвоены текущие дата и время данного компьютера, выраженные как местное время.
Second	Возвращает компонент секунды даты, представленной этим экземпляром.
Ticks	Возвращает число тактов, которое представляет дату и время этого экземпляра.
TimeOfDay	Возвращает время дня для этого экземпляра.
Today	Возвращает текущую дату.
UtcNow	Возвращает объект DateTime, которому присвоены текущие дата и время данного компьютера, выраженные в формате UTC.
Year	Возвращает компонент года даты, представленной этим экземпляром.

5. Напишем запуск таймера, который будет происходить при загрузке формы (двойной щелчок ЛКМ по форме):

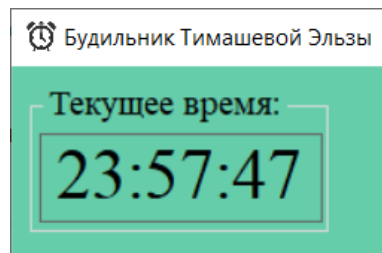
```
private void AlarmForm_Load(object sender, EventArgs e)
{
    //запуск и отображение таймера
    timer.Interval = 1000;
    timer.Tick += new EventHandler(timer_Tick);
    timer.Start();
}
```

Тут `timer.Interval` служит переменной, определяющей промежуток от одного тика до следующего, в миллисекундах, а `EventHandler` создает новое событие, обновляя данные в элементе `timeLabel` на форме.

6. Теперь внесём строчку в обработчик события `OnTimer` нашего таймера (двойной щелчок ЛКМ по таймеру):

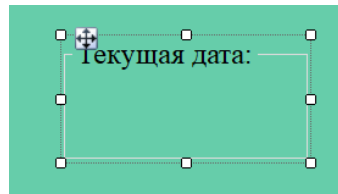
```
private void timer_Tick(object sender, EventArgs e)
{
    timeLabel.Text = DateTime.Now.Hour.ToString("00") + ":"
        + DateTime.Now.Minute.ToString("00") + ":"
        + DateTime.Now.Second.ToString("00");
}
```

Мы получили текущее время и присвоили его свойству `Text` элемента `timeLabel`. Поскольку подобное присваивание будет происходить каждую секунду по событию `OnTimer`, мы получили простейшие электронные часы. Запустить – проверить работу.

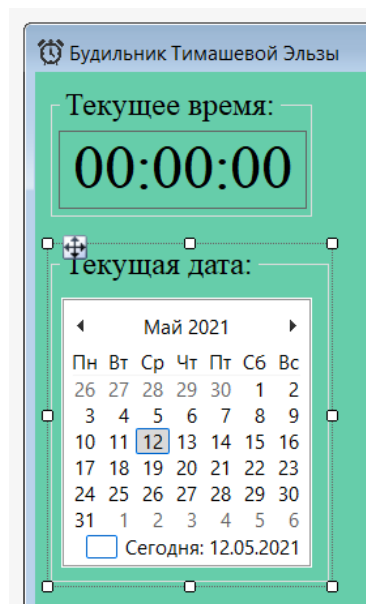


7. Теперь разместим на форме календарь. Для начала установим еще один GroupBox и зададим ему свойства:

- Text – Текущая дата:
- Name – DateBox,



Затем отобразим календарь. Для этого используем элемент MonthCalendar. Установим его в DateBox и подгоним размеры DateBox под него:



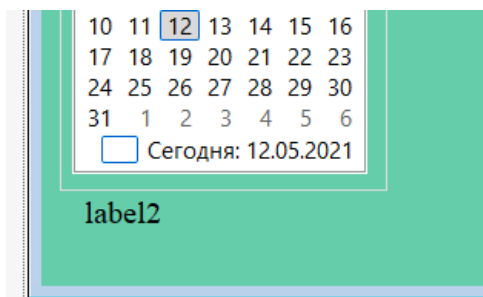
Календарь отражает текущую дату, но не пишет полного названия дня. Исправим это.

Добавим еще один Label на форму, поместив его под календарем.

Зададим ему свойства:

Font – Times New Roman, 12 пт,

Name – dayLabel:

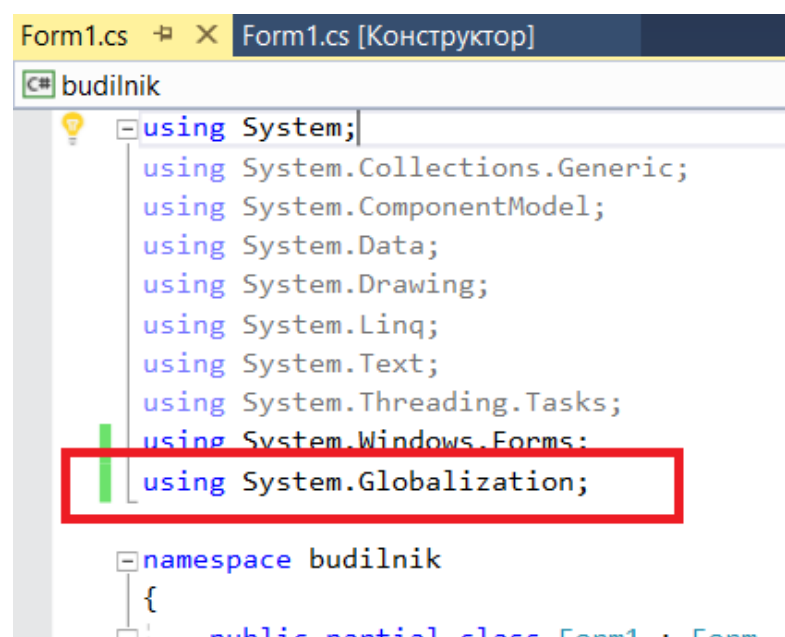


8. Чтобы отобразить текущий день недели, нужно дописать в коде загрузки формы:

```
private void AlarmForm_Load(object sender, EventArgs e)
{
    //запуск и отображение таймера
    timer.Interval = 1000;
    timer.Tick += new EventHandler(timer_Tick);
    timer.Start();

    //отображение дня недели
    dayLabel.Text = String.Format("День недели - {0}",
        CultureInfo.CurrentCulture.DateTimeFormat.GetDayName(DateTime.Now.DayOfWeek));
}
```

Здесь класс **CultureInfo** предоставляет сведения о конкретных языках и региональных параметрах. Эти сведения включают имена языков и региональных параметров, систему письма, используемый календарь, порядок сортировки строк и форматы дат и чисел. Требуется описание пространства имен [System.Globalization](#), добавьте:

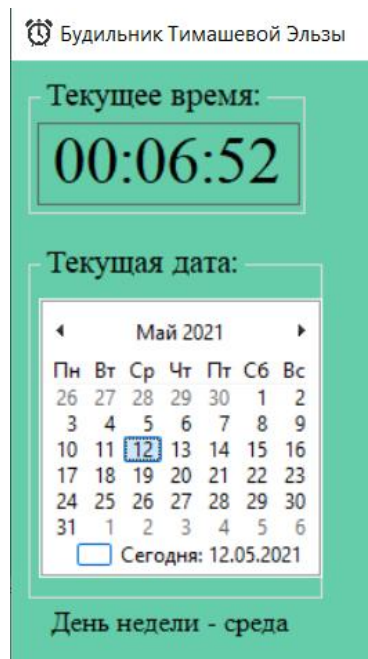


```
Form1.cs [Конструктор]
C# budilnik
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Globalization;

namespace budilnik
{
    public partial class Form1 : Form
```

После запуска получаем текущий день недели и выводим его на русском языке.

Теперь при запуске программы можно увидеть результат проделанной работы:



9. Остается сделать будильник.

Что нам понадобится для этого? Во-первых, какой-нибудь объект, в котором будем задавать значение времени срабатывания будильника (оповещения), медиаплеер для воспроизведения мелодии оповещения, 2 кнопки для задания времени и отмены будильника, и диалог открытия файла для выбора мелодии будильника.

Выделите TimeBox и нажмите клавиши Ctrl+C, чтобы скопировать его в буфер обмена. После этого нажмите клавиши Ctrl+V, чтобы добавить на форму ещё один GroupBox с содержащейся в нём панелью.

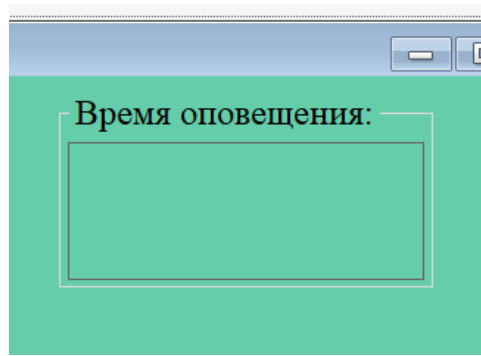
Перетащите новый GroupBox с содержащейся в нём панелью вправо от TimeBox.

(Заметим, что в новых компонентах все внесённые нами изменения свойств сохранились (кроме, естественно, имени – Name, т.к. в программе не может быть двух переменных с одинаковым именем). Заметим также, что способ копировать-вставить очень удобен при создании нескольких объектов с одинаковыми свойствами (например, группы кнопок). Только стоит помнить, что этим приёмом можно пользоваться лишь на этапе проектирования: нельзя копировать кнопки (и вообще компоненты) с уже написанными обработчиками событий, т.к. попытки изменить последние приведут к неработоспособности программы).

10. Задайте следующие значения свойств нового GroupBox:

Text	Время оповещения:
Name	AlarmBox

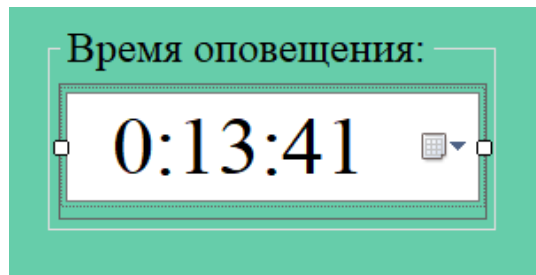
Переименуйте содержащуюся в нём панель (panel1), изменив свойство Name на **AlarmPanel**.



В AlarmBox на элемент AlarmPanel поместим элемент DateTimePicker, задав для него свойства:

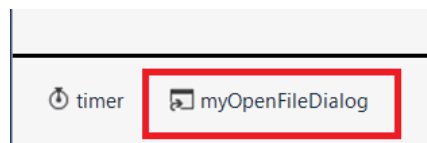
Format = Time

Name = myDateTimePicker



Он понадобится нам для отображения времени звонка.

11. Разместим элемент OpenFileDialog, зададим ему свойство **Name = myOpenFileDialog**:



Поместим на форму также 2 кнопки.

Свойства первой кнопки зададим такими:

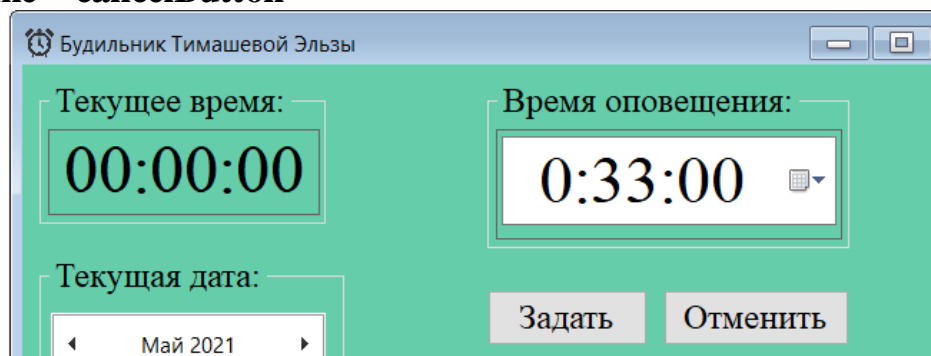
Text – Задать

Name – setButton

У второй:

Text – Отменить

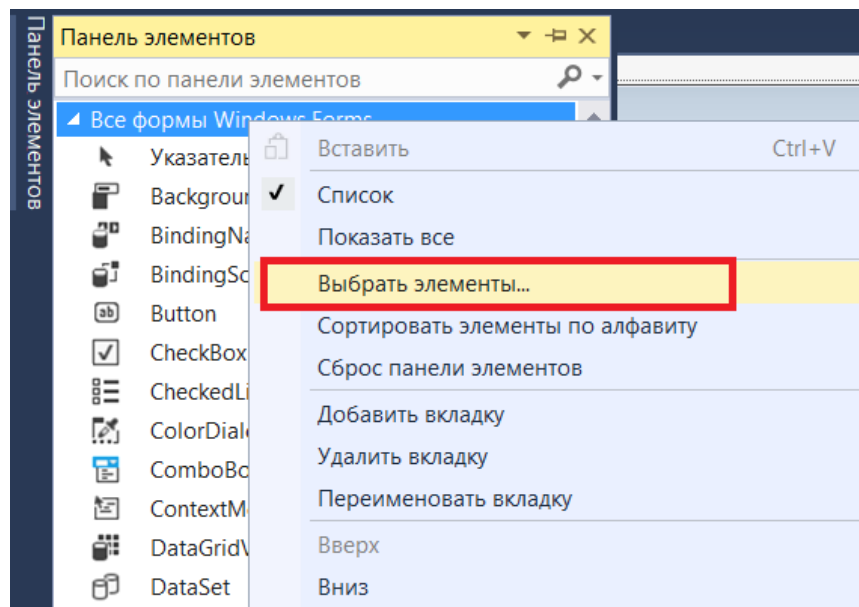
Name – cancelButton



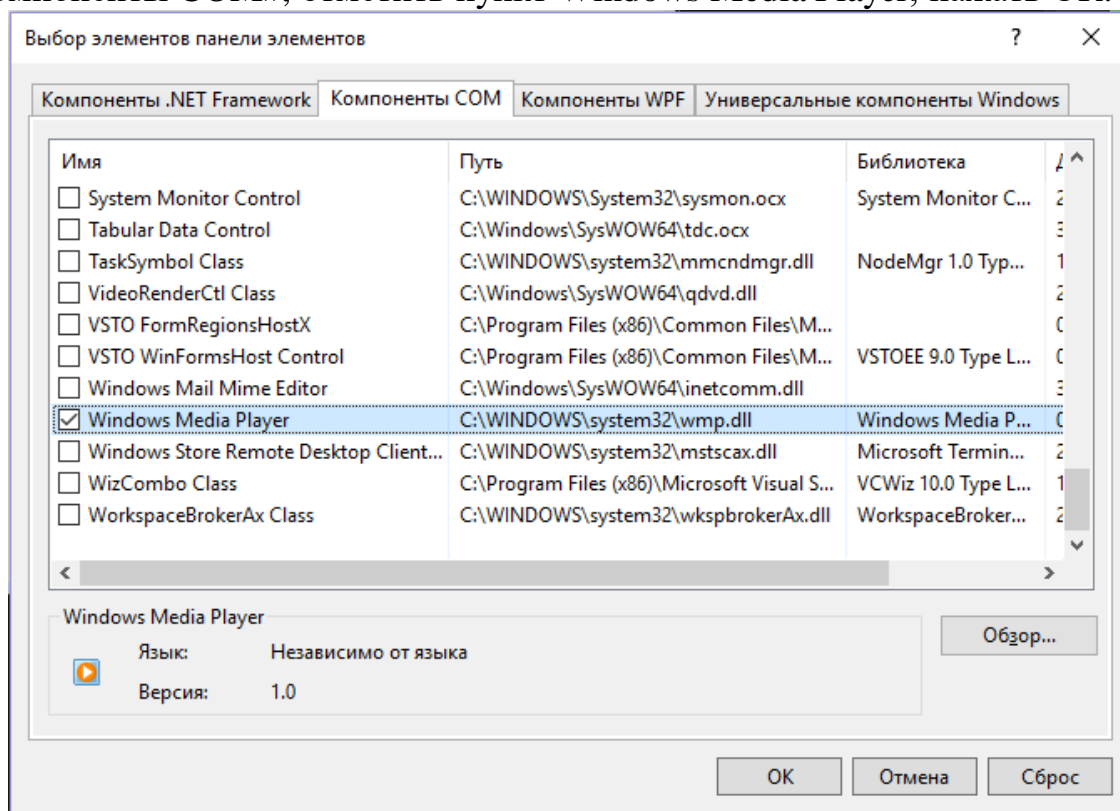
12. Теперь на форму поместим элемент **Panel**.

В него поместим элемент Windows Media Player, однако его не найти в стандартной панели элементов, т.к. он туда не добавлен. Для этого

необходимо: открыть контекстное меню списка «Все формы Windows Forms», который находится на панели элементов, и выбрать пункт «Выбрать элементы...»



После в окне «Выбор элементов панели элементов» открыть вкладку «Компоненты COM», отметить пункт Windows Media Player, нажать ОК.



После этого в списке элементов в панели элементов появится Windows Media Player (если Media Player не появился после добавления, перезапустите среду Visual Studio)

Поместим медиаплеер в Panel.

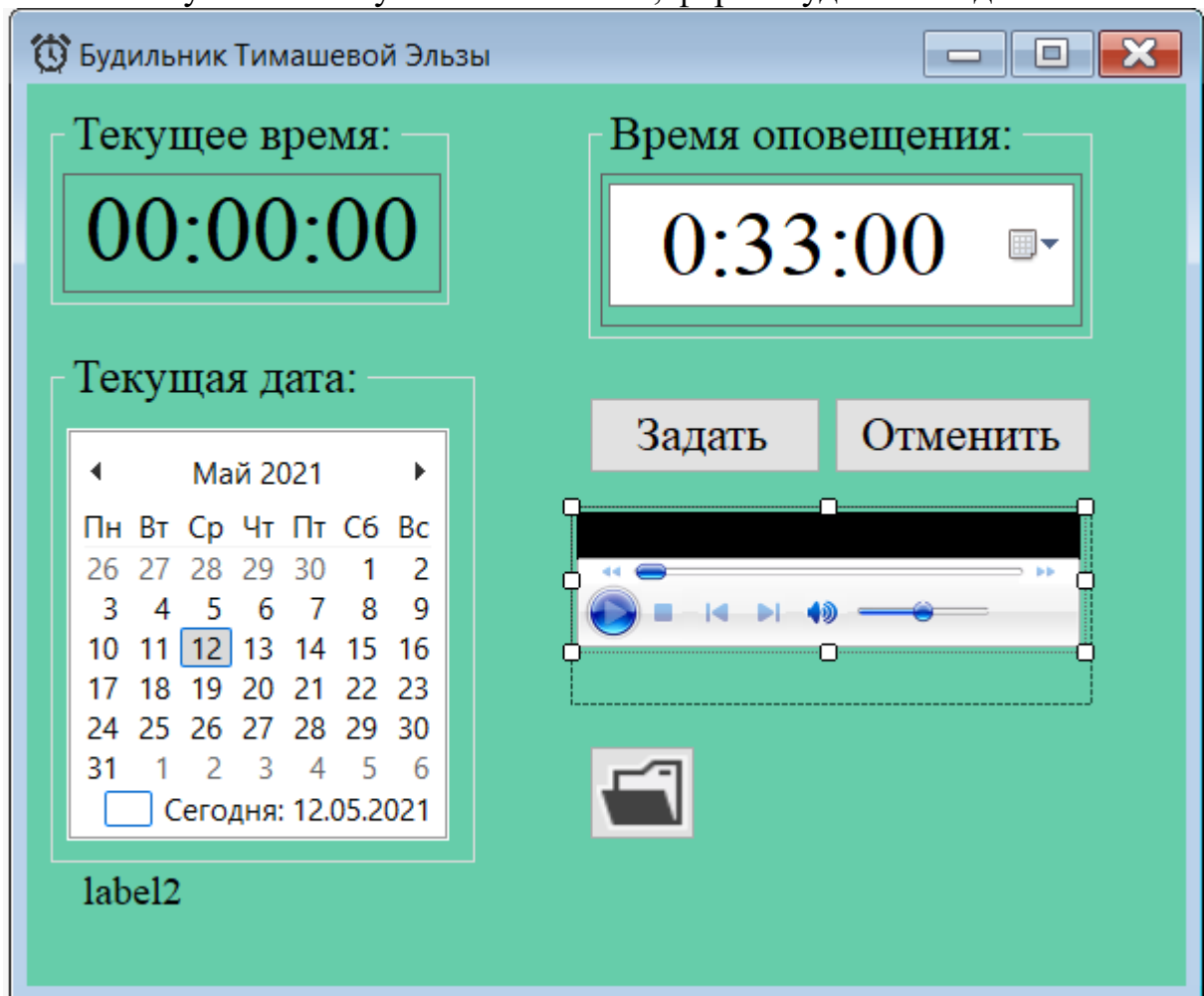
Зададим свойство **Name = myMediaPlayer.**

13. Добавим на форму кнопку для выбора мелодии будильника:

Свойство	Значение
Text	
BackGroundImage	Folder_Open.png
BackGroundImageLayout	Zoom
Size	50;50
Name	setMusicButton



После установки нужных элементов, форма будет выглядеть так:



14. Приступим непосредственно к написанию кода. Нужны переменные для хранения следующих значений: информации о том, включён или выключен будильник (т.е. тип bool), переменная для хранения пути к выбранной мелодии будильника.

Добавьте описание глобальных переменных:

```
bool alarm = false;  
string music = "";
```

В функции AlarmForm_Load сделаем кнопку «Отменить» неактивной, т.к. в ней нет смысла, пока будильник не заведен.

```
private void AlarmForm_Load(object sender, EventArgs e)
{
    //запуск и отображение таймера
    timer.Interval = 1000;
    timer.Tick += new EventHandler(timer_Tick);
    timer.Start();

    //отображение дня недели
    dayLabel.Text = String.Format("День недели - {0}",
        CultureInfo.CurrentCulture.DateTimeFormat.GetDayName(DateTime.Now.DayOfWeek));

    //отключить кнопку Отменить
    if ( !alarm )
    {
        cancelButton.Enabled = false;
    }
}
```

15. Теперь запишем код для кнопок «Задать» и «Отменить»

Код для кнопки «Задать» будет выглядеть так:

```
private void setButton_Click(object sender, EventArgs e)
{
    if (music == "")
        MessageBox.Show("Чтобы задать будильник, назначьте мелодию!");
    else
    {
        if ( !alarm )
        {
            setButton.Enabled = false;
            cancelButton.Enabled = true;
            alarm = true;
        }
    }
}
```

Код кнопки «Отменить»:

```
private void cancelButton_Click(object sender, EventArgs e)
{
    setButton.Enabled = true;
    cancelButton.Enabled = false;
    alarm = false;
    myMediaPlayer.close();
}
```

В самом конце присутствует строка: `myMediaPlayer.close();` которая останавливает музыку после остановки будильника путем закрытия медиаплеера.

16. Код для открытия окна выбора файлов:

```
private void setMusicButton_Click(object sender, EventArgs e)
{
    OpenFileDialog openFile = new OpenFileDialog();
    if (openFile.ShowDialog() == DialogResult.OK)
    {
        music = openFile.FileName;
    }
}
```

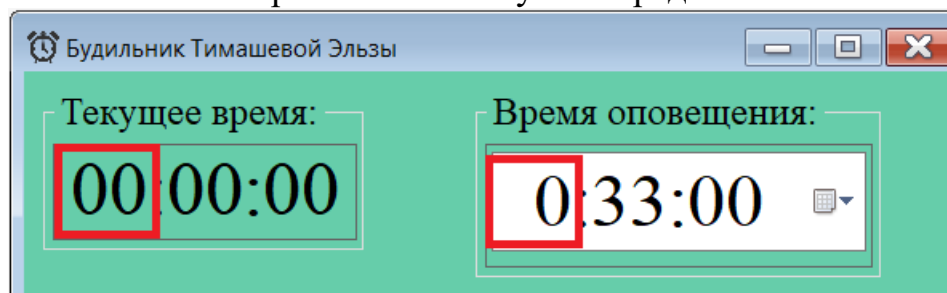
17. Таймер ведет отсчет времени, пока текущее время не совпадет с заданным временем.

После совпадения медиаплеер запустит мелодию. Допишем соответствующие строки кода:

```
private void timer_Tick(object sender, EventArgs e)
{
    //отображение текущего времени
    timeLabel.Text = DateTime.Now.Hour.ToString("00") + ":"
        + DateTime.Now.Minute.ToString("00") + ":"
        + DateTime.Now.Second.ToString("00");

    //работа будильника
    if (Equals(timeLabel.Text, myDateTimePicker.Text))
        myMediaPlayer.URL = music;
}
```

Но этот вариант проверки не сработает в некоторых случаях, а именно в случае, когда время будильника от 0ч0м0с до 9ч59м59с, в связи с тем, что в `DateTimePicker` часы отображаются без нуля впереди:



Поэтому поправим код, добавив проверку на этот случай:

```

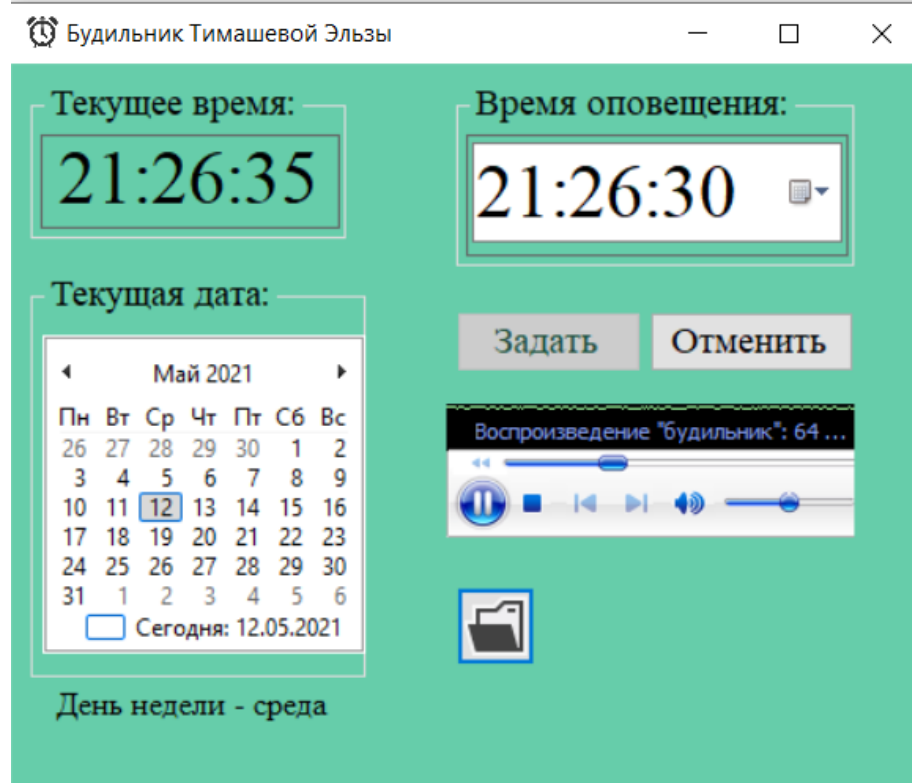
private void timer_Tick(object sender, EventArgs e)
{
    //отображение текущего времени
    timeLabel.Text = DateTime.Now.Hour.ToString("00") + ":"
        + DateTime.Now.Minute.ToString("00") + ":"
        + DateTime.Now.Second.ToString("00");

    //работа будильника
    if (myDateTimePicker.Text.Length == 7)
    {
        if (Equals(timeLabel.Text, "0" + myDateTimePicker.Text))
            myMediaPlayer.URL = music;
    }

    if (Equals(timeLabel.Text, myDateTimePicker.Text))
        myMediaPlayer.URL = music;
}

```

Ну что ж, простейший электронный календарь-часы готов.



Разумеется, эта программа — только «скелет», функциональность которой можно значительно расширить, превратив в полноценный органайзер: добавить поддержку вывода текстовых сообщений-напоминаний в определённый день и час (для этого нужно реализовать запись в файл), небольшую базу данных с таблицами «контакты», «задачи», «встречи» и т.п.