

## Практическое занятие № 28

### Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени.

**Цель:** получить навыки управления данными даты и времени.

**Норма времени:** 2 часа.

**Оборудование:** Компьютер, среда программирования Visual Studio.

### Порядок выполнения работы

#### Задания:

1. Изучить теоретическую часть.
2. Выполнить задания, следуя указаниям.
3. Ответить на контрольные вопросы.
4. Предъявить преподавателю результаты работы программы и исходные коды.

### Теоретическая часть

Для работы с датами в **Windows Forms** имеются элементы **DateTimePicker** и **MonthCalendar**.

Использование элемента управления, отображающего календарь, значительно упрощает для пользователя выбор даты. Кроме того, такие элементы управления гарантируют, что дата будет отформатирована правильно. Календарь можно отобразить с помощью элемента управления *MonthCalendar* или *DateTimePicker*.

Элемент управления *MonthCalendar* позволяет отображать календарь для одного или нескольких месяцев. При этом пользователи могут выбирать отдельную дату или диапазон дат.

Элемент управления *DateTimePicker* имеет два состояния. По умолчанию элемент управления *DateTimePicker* выглядит как текстовое поле с раскрывающимся списком в виде стрелки. Когда пользователь нажимает на стрелку раскрывающегося списка, появляется календарь. При использовании этого элемента управления пользователь может выбрать только одну дату. Элемент управления *DateTimePicker* также позволяет отображать время вместо дат.

Процесс, используемый для извлечения даты из этих элементов управления, зависит от конкретного используемого элемента. Используйте свойство *Start* для элемента управления *MonthCalendar* и свойство *Value* для элемента управления *DateTimePicker*.

Элемент управления *MonthCalendar* позволяет отображать на экране одновременно до 12 месяцев. По умолчанию в этом элементе управления отображается только один месяц, однако имеется возможность указать количество месяцев, которые будут отображаться на экране, и их размещение в данном элементе управления. Чтобы обеспечить достаточное количество места в форме для новой размерности, при изменении диапазона календаря изменяются размеры элемента управления.

#### Константы для указания формата даты

Константа	Описание	Пример
DateFormat.GeneralDate	Отображает дату, время или оба значения. Если присутствует дата, она отображается в кратком формате. Если присутствует время, оно отображается в полном формате. Если присутствует и время, и дата, отображаются обе части.	22/11/1963 12:00:00 PM
DateFormat.LongDate	Отображает дату в полном формате, который	Пятница, 22

	определяется установленными на компьютере региональными параметрами.	ноября, 1963
DateFormat.ShortDate	Отображает дату в кратком формате, который определяется установленными на компьютере региональными параметрами.	11/22/1963
DateFormat.LongTime	Отображает время в полном формате, который определяется установленными на компьютере региональными параметрами.	12:00:00 PM
DateFormat.ShortTime	Отображает время в 24-часовом формате (чч:мм).	12:00

### Свойства и функции системных часов

Чтобы получить от системных часов информацию о времени, можно использовать их различные свойства и функции. Информация о времени может потребоваться в программах при создании собственных календарей, часов или оповещений. В следующей таблице содержится перечень наиболее полезных функций системных часов. За дополнительной информацией обращайтесь к справочной системе VisualStudio.

Свойство или функция	Описание
TimeString	Возвращает от системных часов текущее время.
DateString	Возвращает от системных часов текущую дату.
Now	Возвращает закодированное значение, содержащее текущие дату и время. Наиболее полезно как аргумент для других функций системных часов.
Hour (time)	Возвращает количество часов для указанного времени (от 0 до 24).
Minute (time)	Возвращает количество минут для указанного времени (от 0 до 59).
Second (time)	Возвращает количество секунд для указанного времени (от 0 до 59).
Day (date)	Возвращает целое число, представляющее собой день месяца (от 1 до 31).
Month (date)	Возвращает целое число, представляющее собой месяц (от 1 до 12).
Year (date)	Возвращает год для указанной даты.
Weekday (date)	Возвращает целое число, представляющее собой день недели (по американской системе: 1 - это воскресенье, 2 - это понедельник, и т.д.).

### Задание на практическое занятие:

#### Порядок выполнения работы

1. Создать новый проект командой *Создать проект (New Project)* из меню *Файл (File)* (порядок создания нового проекта подробно описан в лабораторной работе № 1).
2. Выберите элемент Приложение WindowsForms и нажмите кнопку ОК.
3. Добавьте в форму элемент *Label*, оставив имя по умолчанию *Label1*.
4. Удалите текст из свойства *Text* элемента управления *Метка*.
5. Добавьте в форму элемент управления *MonthCalendar*, оставив имя по умолчанию *MonthCalendar1*.
6. Дважды щелкните элемент управления *MonthCalendar*, чтобы открыть обработчик событий по умолчанию в редакторе кода.
7. В обработчике событий *MonthCalendar1\_DateChanged* добавьте следующий код для добавления элементов в список.

```
Me.Label1.Text = CStr(Me.MonthCalendar1.SelectionRange.Start)
```

8. Вернитесь в режим конструктора и добавьте в форму элемент управления *DateTimePicker*, оставив имя по умолчанию *DateTimePicker1*.

9. Дважды щелкните элемент управления *DateTimePicker*, чтобы открыть обработчик событий по умолчанию в редакторе кода.

10. В обработчике событий *DateTimePicker\_ValueChanged* добавьте следующий код для добавления элементов в список.

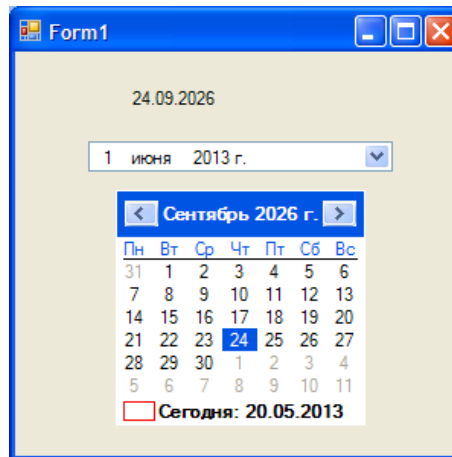
```
Me.Label1.Text = CStr(Me.DateTimePicker1.Value)
```

11. Нажмите клавишу F5 для запуска программы.

12. Когда появится форма, выберите дату в элементе управления *MonthCalendar* и убедитесь, что она отображается в метке.

13. Щелкните стрелку раскрывающегося списка элемента управления *DateTimePicker* и выберите дату.

Дата и время отображаются в метке.



### Извлечение нескольких дат

Диапазон дат, выбранных в элементе управления *MonthCalendar*, можно извлечь с помощью свойств *Start* и *End* свойства *SelectionRange*. По умолчанию максимальное число дней, которые можно выбрать, равно 7, но при необходимости этот параметр можно изменить, установив значение свойства *MaxSelectionCount*. Чтобы определить, выбран ли диапазон дат, просто проверьте, совпадают ли даты начала и конца.

### Извлечение диапазона дат из элемента управления календарем месяца

1. Замените код в обработчике событий *MonthCalendar1\_DateChanged* следующим. Этот код устанавливает максимальное число дней (две недели), которые могут быть выбраны в элементе управления. Он отображает дату начала в метке, если выбран только один день, и отображает диапазон дат при выборе диапазона дней в элементе управления *MonthCalendar*.

```
Me.MonthCalendar1.MaxSelectionCount = 14

If Me.MonthCalendar1.SelectionRange.Start = _
    Me.MonthCalendar1.SelectionRange.End Then

    Me.Label1.Text = CStr(Me.MonthCalendar1.SelectionStart)

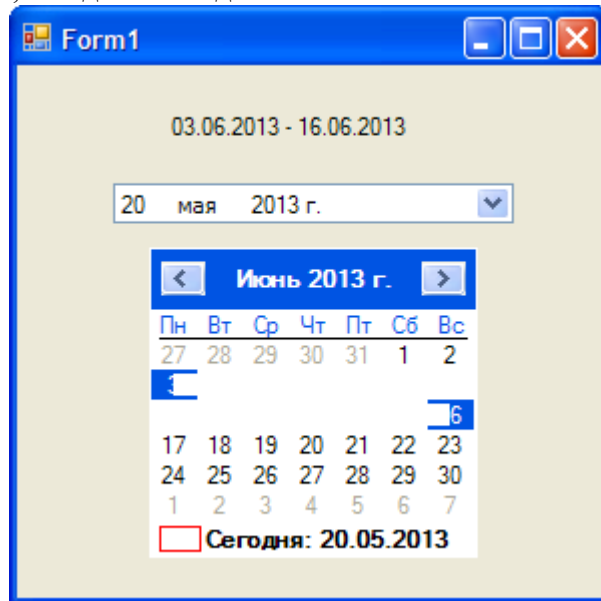
Else

    Me.Label1.Text = Me.MonthCalendar1.SelectionRange.Start & _
        " - " & Me.MonthCalendar1.SelectionRange.End

End If
```

2. Нажмите клавишу F5 для запуска программы.

3. Когда появится форма, выберите диапазон дат в элементе управления *MonthCalendar* и убедитесь, что диапазон дат появился в метке.



### Форматирование дат

Даты, возвращаемые элементами управления *MonthCalendar* и *DateTimePicker*, можно форматировать с помощью функции *FormatDateTime*. Существует несколько констант, которые можно использовать для указания формата даты (см. теор. часть).

### Форматирование даты в метке

1. Замените код в обработчике событий *MonthCalendar1\_DateChanged* следующим. Этот код форматирует дату, возвращаемую в полном формате.

```
Me.MonthCalendar1.MaxSelectionCount = 14
```

```
If Me.MonthCalendar1.SelectionRange.Start = _  
    Me.MonthCalendar1.SelectionRange.End Then
```

```
    Me.Label1.Text = FormatDateTime( _  
        Me.MonthCalendar1.SelectionStart, _  
        DateFormat.LongDate)
```

```
Else
```

```
    Me.Label1.Text = FormatDateTime( _  
        Me.MonthCalendar1.SelectionRange.Start, _  
        DateFormat.LongDate) & " - " & FormatDateTime( _  
        Me.MonthCalendar1.SelectionRange.End, DateFormat.LongDate)
```

```
End If
```

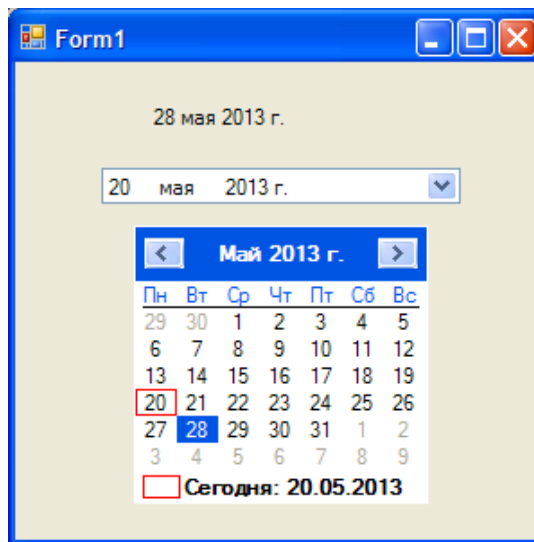
2. Замените код в обработчике событий *DateTimePicker1\_ValueChanged* следующим. Этот код форматирует дату, возвращаемую в полном формате.

```
Me.Label1.Text = FormatDateTime(Me.DateTimePicker1.Value, _  
    DateFormat.LongDate)
```

3. Нажмите клавишу F5 для запуска программы.

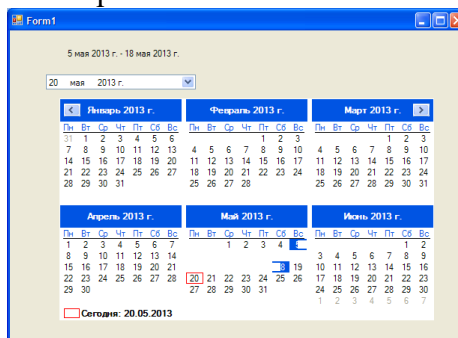
4. Когда появится форма, выберите дату или диапазон дат в элементе управления *MonthCalendar*. Убедитесь, что дата или диапазон дат отображается в метке в полном формате.

5. Выберите дату в элементе управления *DateTimePicker* и убедитесь, что дата в метке отображается в полном формате.



### Чтобы отобразить несколько месяцев

- Задайте для свойства *CalendarDimensions* значение, равное числу месяцев, отображаемых по горизонтали и вертикали.



```
MonthCalendar1.CalendarDimensions = New System.Drawing.Size(3, 2)
```

### Программа Birthday

В программе Birthday элементы управления *DateTimePicker* и *Button* используются, чтобы выяснить у пользователя дату его рождения и показать эту информацию в окне сообщения.

1. В области элементов выберите элемент управления *Button*, и ниже объекта выбора даты и времени добавьте объект кнопки. Эта кнопка будет использована для показа дня рождения и для проверки правильности работы объекта выбора даты и времени.

2. В окне *Свойства(Properties)* измените свойство *Text* объекта кнопки на **Показать день моего рождения.**

3. Дважды щелкните мышью на объекте кнопки, а потом наберите следующий фрагмент программы между операторами *Private Sub* и *End Sub* в процедуре события *Button1\_Click*:

```
MsgBox("Ваш день рождения " & DateTimePicker1.Text)
MsgBox("День года: " & DateTimePicker1.Value.DayOfYear.ToString())
MsgBox("Сейчас: " & DateTimePicker1.Value.TimeOfDay.ToString())
```

Этот фрагмент программы показывает три последовательных окна сообщения (небольшие диалоговые окна), которые содержат информацию из объекта календаря. В первой строке используется свойство *Text* календаря для вывода информации о дате рождения, которую пользователь выберет в этом объекте после запуска программы. Функция *MsgBox* кроме текстового значения из свойства *Text* календаря показывает

строку "Ваш день рождения". Эти два текстовых элемента объединяются в строку с помощью оператора конкатенации (слияния) строк &.

Во второй строке `DateTimePicker1.Value.DayOfYear.ToString()` объект календаря используется для вычисления дня года, отсчитывая с 1 января. Это делается с помощью свойства `DayOfYear` и метода `ToString`, который переводит числовой результат вычисления даты в текстовое значение, которое гораздо проще показать с помощью функции `MsgBox`.

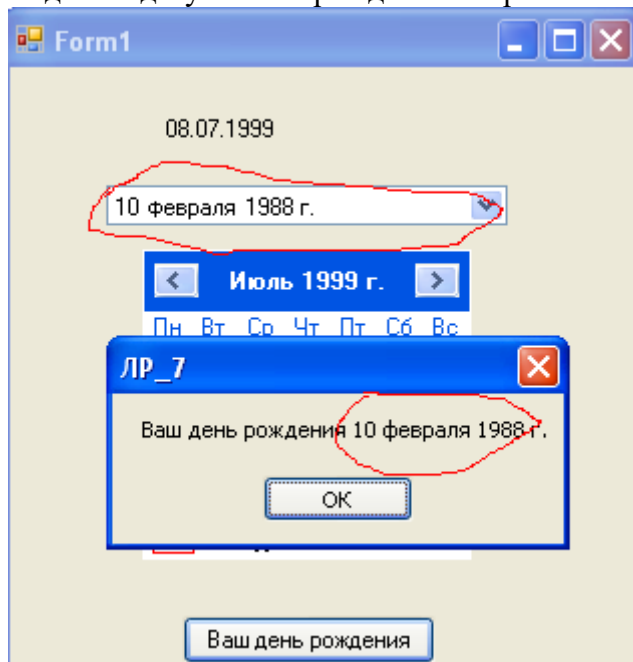
В третьей строке фрагмента, после перевода значения в строковое (или текстовое) представление, в окне сообщения показывается информация о точном времени.

### Запуск программы Birthday

1. На стандартной панели инструментов нажмите кнопку Start (Начать). Программа Birthday запустится в среде разработки. В окне объекта выбора даты и времени появится текущая дата.

2. Нажмите стрелку раскрывающегося списка, чтобы вывести на экран представление этого объекта в виде календаря. Форма будет выглядеть как на следующей иллюстрации.

3. Выберите в элементе `DateTimePicker1` число, месяц и год Вашего рождения, пользуясь стрелками прокрутки. Нажмите кнопку Показать день моего рождения. Visual Basic исполнит введенный вами код программы и покажет окно с сообщением, содержащим день и дату вашего рождения. Обратите внимание на соответствие двух дат.



4. В окне сообщения нажмите ОК. Появится второе окно сообщения, указывающее, в какой день года вы родились.

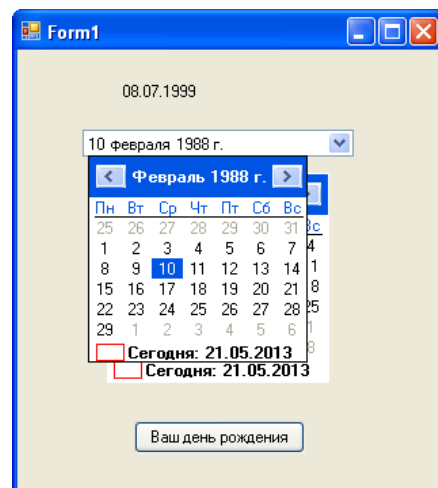
5. Нажмите ОК, чтобы показать последнее окно сообщения. Появятся текущие дата и время. Вы обнаружите, что объект выбора даты и времени очень удобен - он не только помнит новую, введенную вами информацию о дате или времени, но также отслеживает текущие дату и время и может показывать эту информацию в различных форматах.

**Совет.** Чтобы настроить объект выбора даты и времени для показа времени, а не даты, установите свойство `Format` этого объекта равным `Time`.


### Работа с датами в VB.Net. (Дополнительно)

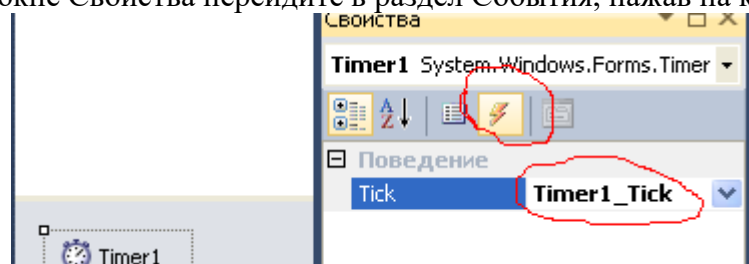
Реализовать программу которая будет узнавать текущую дату и время.

Чтобы узнать текущее время в VB.Net есть функция - **TimeString**, чтобы узнать дату - **DateString**. Кроме того есть функции работы со временем:



**Hour()**(часы), **Minute()**(минуты), **Second()**(секунды). Эти функции вырезают часы, минуты, секунды из указанно времени. Например, *Minute(TimeString)* - вырезает минуты из текущего времени. И функции для работы с датами: **Day()**(год), **Month()**(месяц), **Year()**(год). Эти функции вырезают из текущей даты: день, месяц, год. Например, *Month(DateString)*, вырезает из текущей даты месяц.

1. Создайте новый проект командой *Создать проект* из меню *Файл (File)*. Выберите элемент Приложение Windows Forms и нажмите кнопку ОК.
2. Добавьте в форму три элемента *Label*.
3. Чтобы размеры метки автоматически регулировались, в зависимости от текста в метке Задайте *AutoSize = True*.
4. Удалите текст из свойства *Text* элемента управления Метка: *Text = ""*.
5. Добавьте в форму невидуемый элемент управления *Timer(Enabled = True, Interval = 1000)*.
6. Напишите обработчик события *Tick* элемента *Timer*. Для этого выделите Элемент Таймер и в окне Свойства перейдите в раздел События, нажав на кнопку .



7. Дважды щелкните в строке **Timer1\_Tick** и введите в открывшемся окне код:

```
Label1.Text = "Время: " & TimeString  
Label2.Text = "Дата: " & DateString
```

8. Для определения дня недели нужна функция **WeekDay**. Напишите обработчик события загрузки формы:

```
Dim Den As Integer  
Den = Weekday(Today) ' Определяем день недели  
If Den = 1 Then Label3.Text = "Воскресенье"  
If Den = 2 Then Label3.Text = "Понедельник"  
If Den = 3 Then Label3.Text = "Вторник"  
If Den = 4 Then Label3.Text = "Среда"  
If Den = 5 Then Label3.Text = "Четверг"  
If Den = 6 Then Label3.Text = "Пятница"  
If Den = 0 Then Label3.Text = "Суббота"
```

9. В области элементов выберите элемент управления *Button* и добавьте его, задайте свойство *Text = Текущая дата/Время*

10. Напишите обработчик события нажатия кнопки для вывода текущей даты или времени:

```

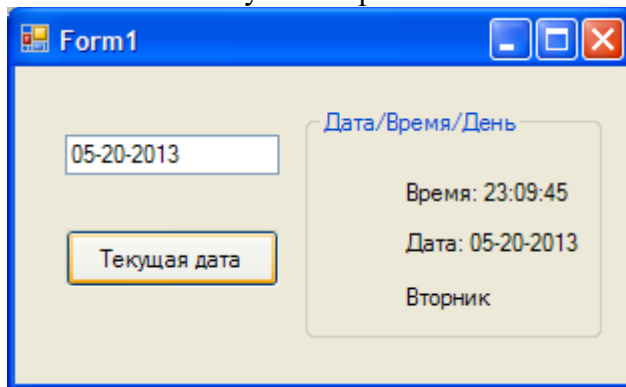
'получить текущие дату и время
TextBox1.Text = My.Computer.Clock.LocalTime ' (microsoft.viual.basic)
TextBox1.Text = DateTime.Now ' (microsoft.viual.basic)
TextBox1.Text = DateTime.Now ' (mscorlib)

'получить только текущее время
TextBox1.Text = DateTime.Now.ToLongTimeString
TextBox1.Text = DateTime.Now.ToLongTimeString
TextBox1.Text = Format(Now, "hh:mm:ss")
TextBox1.Text = TimeString

'получить только текущую дату
TextBox1.Text = DateTime.Now.Date
TextBox1.Text = DateTime.Today
TextBox1.Text = DateTime.Now.Date
TextBox1.Text = DateTime.Today
TextBox1.Text = Format(Now, "dd.MM.yyyy")
TextBox1.Text = DateString

```

11. Запустите проект.



### Контрольные вопросы

1. Назначение элемента управления DateTimePicker
2. Назначение элемента управления MonthCalendar
3. Назовите свойства, используемые для извлечения даты из этих элементов.
4. Свойства элемента управления MonthCalendar для извлечения диапазона дат.
5. Назначение оператора конкатенации &, пример его использования
6. Отображение нескольких месяцев элемента управления MonthCalendar.
7. Функция форматирования дат.