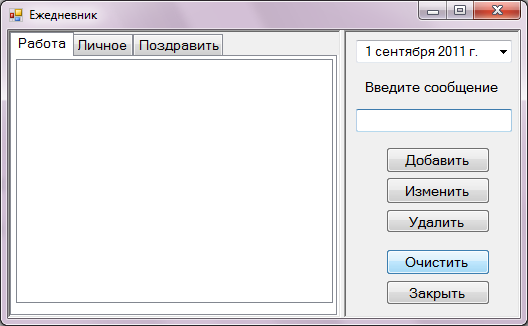
## Лабораторная работа № 8. «Ежедневник»

**Задание.** Разработайте приложение «Ежедневник», позволяющее создавать, редактировать, удалять, а также сохранять в файле и загружать из файла записи на любой календарный день (рис. 8.1).



*Рис. 8.1.* Приложение «Ежедневник»

**Решение**

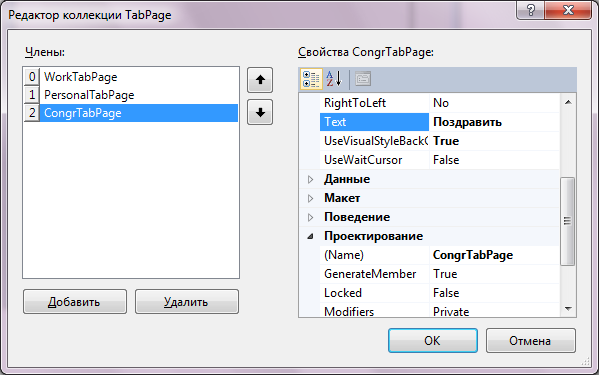
Создайте новый проект *Organizer* в решении *Lab8*.

**1‑й этап:** визуальное проектирование. Измените заголовок формы на «Ежедневник», значение свойства *Name* на *MainForm*, размеры формы (*Size*) установите равными 700; 400.

Поместите на форму компонент SplitContainer, позволяющий разделить рабочую область приложения на две части. Измените значения свойств компонента следующим образом:

|  |  |
| --- | --- |
| Свойство | Значение |
| *BorderStyle* | *Fixed3D* |
| *Dock* | *Fill* |
| *SplitterDistance* | 450 |

На левую панель компонента SplitContainer поместите элемент уп­рав­ления TabControl, переименуйте его в *OrgTabControl*. Создайте вкладки данного компонента, для этого в окне Свойств найдите свойство знак *TabPages*, щелкните по кнопке с тремя точками в правой части второго столбца для открытия диалогового окна редактирования значений выбран­ного свойства.



*Рис. 8.2.* Создание вкладок для компонента *OrgTabControl*

В появившемсядиалоговом окне (рис. 8.2) необходимо добавить к двум существующим по умолчанию еще одну вкладку (с помощью кнопки *Добавить*), а затем установить значения свойств каждой вкладки следующим образом:

|  |  |  |
| --- | --- | --- |
| Компонент | Свойство | Значение |
| *tabPage1* | *Name* | *WorkTabPage* |
| *Text* | Работа |
| *tabPage2* | *Name* | *PersonalTabPage* |
| *Text* | Личное |
| *tabPage3* | *Name* | *CongrTabPage* |
| *Text* | Поздравить |

Закройте диалоговое окно, нажав на кнопку *OK*.

Положите на каждую вкладку ежедневника *OrgTabControl* компонент ListBox (список строк) для размещения заметок пользователя. Для этого выполните такую последовательность действий. Выберите вкладку «Работа» и щелкните внутри нее (при этом в окне Свойств будет выбран компонент *WorkTabPage*), в палитре компонентов выберите компонент ListBox и поместите его на текущую вкладку. Далее свойство *Dock* компонента *listBox1* установите в *Fill*. Скопируйте компонент *listBox1* в буфер обмена (щелк­ните по объекту *listBox1* и выполните команду *Правка* – *Копировать*).

На вторую страницу ежедневника поместите компонент *listBox2*. Для этого выберите вкладку «Личное» и щелкните внутри нее. Поместите копию компонента ListBox на страницу ежедневника, выполнив команду *Правка – Вставить*. Повторите эту операцию для третьей страницы ежедневника.

Свойство *Dock* компонента *OrgTabControl* установите в *Fill*.

На правую панель компонента SplitContainer положите один компонент Label, один компонент TextBox и пять компонентов Button, расположите их так, как показано на рис. 8.1.

Установите свойства компонентов следующим образом:

|  |  |  |
| --- | --- | --- |
| Компонент | Свойство | Значение |
| *Label1* | *Text* | Введите сообщение |
| *TextBox* | *Name* | *RecordTextBox* |
| *Text* |  |
| *Button1* | *Name* | *AddButton* |
| *Text* | Добавить |
| *Button2* | *Name* | *ChangeButton* |
| *Text* | Изменить |
| *Button3* | *Name* | *DeleteButton* |
| *Text* | Удалить |
| *Button4* | *Name* | *ClearButton* |
| *Text* | Очистить |
| *Button5* | *Name* | *CloseButton* |
| *Text* | Закрыть |

Для выбора даты, для которой будут отображаться записи, воспользуемся элементом управления **DateTimePicker** (). Он позволяет выбрать отдельный элемент в списке дат или времени. Когда компонент используется для представления даты, он состоит из двух частей: раскрывающегося списка с датой, представленной в виде текста, и сетки календаря, которая появляется при нажатии кнопки со стрелкой вниз, расположенной рядом со списком.

В табл. 8.1 перечислены некоторые свойства элемента управления DateTimePicker.

*Таблица 8.1*

**Некоторые свойства элемента управления DateTimePicker**

|  |  |
| --- | --- |
| Свойство | Описание |
| *CalendarFont* | Возвращает или задает стиль шрифта, применяемый к календарю |
| *CalendarForeColor, CalendarMonthBackground* | Возвращает или задает основной цвет/цвет фона календаря |
| *Format* | Определяет формат даты и времени, отображаемых в элементе управления |
| *MaxDate, MinDate* | Задает максимальное/минимальное значение даты и времени, которые могут быть выбраны в элементе управления |
| *Text* | Получает или задает текст, сопоставленный с этим элементом управления |
| *Value* | Возвращает или задает значение даты/времени, назначаемое элементу управления |

При изменении значения свойства *Value* элемента управления Date­Time­Picker происходит событие ValueChanged.

На правую панель компонента SplitContainer положите один компонент DateTimePicker (см. рис. 8.1).

Установите значение свойства *Font.Size* всех компонентов на форме равным 10. Расположите компоненты красиво.

Итак, визуальное проектирование главной формы приложения завершено.

**2‑й этап.** Написание программного кода.

После щелчка по кнопке «Добавить» содержимое окна редактирования (*RecordTextBox*) будет добавляться в список заметок на текущей странице ежедневника. Для операций со строками у компонента ListBox имеется свойство *Items*. Чтобы добавить строку в список, используется метод *Add*(). Создайте обработчик события Click для кнопки *AddButton*.

***1‑й способ***

Обработчик события будет выглядеть следующим образом:

private void AddButton\_Click(object sender, EventArgs e)

{

switch (OrgTabControl.SelectedIndex)

{

case 0: listBox1.Items.Add(RecordTextBox.Text);

break;

case 1: listBox2.Items.Add(RecordTextBox.Text);

break;

case 2: listBox3.Items.Add(RecordTextBox.Text);

break;

}

RecordTextBox.Text = "";

}

Здесь в зависимости от индекса текущей вкладки ежедневника (нумерация с нуля), значение которой возвращает свойство *SelectedIndex* компонента *OrgTabControl*, добавляем запись в соответствующий этой странице компонент ListBoх (нумерация с единицы) с помощью метода *Add*().

Запустите приложение. Попробуйте добавить строки на разные страницы ежедневника.

При большом количестве страниц подобный код был бы слишком громоздким. Тогда можно было бы написать процедуру другим способом.

***2‑й способ***

В *С*# для компонента Form определено свойство *Controls*, которое получает коллекцию всех элементов управления, содержащихся на форме. Для этого свойства определен метод *Find*(), выполняющий поиск элементов управления по их свойству [*Name*](ms-help://MS.VSCC.v90/MS.MSDNQTR.v90.ru/fxref_system.windows.forms/html/ac7ade9f-4d18-bc74-e335-83e4e453f3af.htm) и создающий массив из всех элементов управления, которые соответствуют условиям поиска. Воспользуемся этим методом в коде обработчика события Click кнопки *AddButton*.

private void AddButton\_Click(object sender, EventArgs e)

{

//определение выбранного компонента ListBox

string s = "listBox" + (OrgTabControl.SelectedIndex + 1);

ListBox CurrentListBox = (ListBox)Controls.Find(s, true)[0];

//добавление записи на текущий ListBox

CurrentListBox.Items.Add(RecordTextBox.Text);

//очистка окна ввода

RecordTextBox.Text = "";

}

Запустите проект. Попробуйте добавить строки на разные страницы ежедневника.

При таком способе можно в любой момент разработки добавлять новые страницы ежедневника (например, «Важное» или «Купить») и это не повлечет за собой необходимости изменения программного кода.

Воспользуемся этим же способом определения компонента ListBox, расположенного на текущей вкладке ежедневника, для того чтобы при выборе определенной строки в списке заметок пользователя в окне редактирования *RecordTextBox* отображалась выбранная запись. Создайте обработчик события Click для компонента *listBox1*.

private void listBox1\_Click(object sender, EventArgs e)

{

//определение выбранного компонента ListBox

string s = "listBox" + (OrgTabControl.SelectedIndex + 1);

ListBox CurrentListBox = (ListBox)Controls.Find(s, true)[0];

//запись в окно редактирования выбранного значения

RecordTextBox.Text = (string)CurrentListBox.SelectedItem;

}

Далее для всех остальных компонентов ListBox укажите, чтобы при событии Click выполнялся обработчик этого события компонента *listBox1*. Для этого для каждого из оставшихся компонентов ListBox в окне Свойств на вкладке событий справа от события Click выберите в списке *listBox1\_Click*.

Запустите проект. Проверьте правильность выполнения.

Для того чтобы заметки, введенные пользователем, сохранялись и после закрытия приложения и загружались в ежедневник при его последующем открытии, необходимо организовать их сохранение в файлы и считывание из файлов. Положим, что содержимое каждого дня ежедневника будет храниться в отдельном текстовом файле с расширением “*org*”, при этом название файла будет совпадать с выбранной пользователем датой, например *«2 октября 2014 г.org»*.

Для хранения имени текущего файла опишите в классе формы строковую переменную *FileName*. В обработчике события Load главной формы приложения *MainForm* определите значение этой переменной:

private void MainForm\_Load(object sender, EventArgs e)

{

FileName = dateTimePicker1.Text + "org";

}

Опишите метод *SaveToFile*() сохранения содержимого ежедневника в файл с заданным именем следующим образом:

private void SaveToFile(string FileName)

{

try

{

//открываем файл для записи

using (StreamWriter sw = new StreamWriter(FileName))

{

//перебираем все компоненты ListBox

for (int i = 1; i <= 3; i++)

{

//задаем текущий компонент ListBox

ListBox CurListBox =

(ListBox)Controls.Find("listBox" + i, true)[0];

//записываем в файл кол‑во строк списка

sw.WriteLine(CurListBox.Items.Count.ToString());

//записываем в файл все записи из списка

for (int j = 0; j < CurListBox.Items.Count; j++)

sw.WriteLine(CurListBox.Items[j]);

//очищаем список записей текущего ListBox

CurListBox.Items.Clear();

}

}

}

catch

{

MessageBox.Show("Ошибка при сохранении!");

}

}

Поясним код. Для записи данных в текстовый файл, имя которого определено в переменной *FileName*, создаем экземпляр класса *StreamWriter* (не забудьте подключить *System.IO* в разделе *using*). Далее последовательно записываем в файл информацию, содержащуюся в каждом из трех компонентов ListBox, предварительно указав количество записей в текущем списке. В результате файл будет иметь следующую структуру:

*N*1 – количество строк в компоненте *listBox1*

*listBox1*.*Items*[1]

… строки, содержащиеся в списке *listBox1*

*listBox1*.*Items*[*N*1]

*N*2 – количество строк в компоненте *listBox2*

*listBox2*.*Items*[1]

… строки, содержащиеся в списке *listBox2*

*listBox2*.*Items*[*N*2]

*N*3 – количество строк в компоненте *listBox3*

*listBox3*.*Items*[1]

… строки, содержащиеся в списке *listBox3*

*listBox3*.*Items*[*N*3]

Отметим, что в случае, когда при сохранении данных указывается короткое имя файла (без указания пути), файл сохраняется в каталог «…/*bin*/ *Debug*» текущего решения.

Аналогичным образом опишите метод *LoadFromFile*() считывания данных из заданного файла. Обратите внимание на то, что в случае, когда заданный файл не найден (блок *catch*), нужно полностью очищать содержимое страниц ежедневника.

Итак, при запуске приложения необходимо загрузить данные из соответствующего файла. Добавьте в конец обработчика события Load главной формы приложения *MainForm* вызов метода LoadFromFile():

private void MainForm\_Load(object sender, EventArgs e)

{

FileName = dateTimePicker1.Text + "org";

LoadFromFile(FileName);

}

При выборе некоторой даты в компоненте DateTimePicker необходимо, во‑первых, сохранить содержимое списков записей текущего дня, далее изменить дату и загрузить содержимое списков записей из соответствующего файла. В случае, если такой файл не найден, необходимо просто очистить содержимое страниц ежедневника.

Создайте обработчик события ValueChanged компонента DateTimePicker и поместите туда следующий программный код:

private void dateTimePicker1\_ValueChanged

(object sender, EventArgs e)

{

SaveToFile(FileName);

FileName = dateTimePicker1.Text + "org";

LoadFromFile(FileName);

}

Запустите приложение, убедитесь в его работоспособности.

**Задания для самостоятельного выполнения**

* 1. Написать обработчик события нажатия на кнопку «Изменить»: текущая строка в списке заметок пользователя должна быть заменена содержимым окна редактирования.
  2. Добавьте в ежедневник вкладку «Купить», обеспечьте корректность работы с записями на ней.
  3. Написать обработчик события нажатия на кнопку «Удалить»: текущая строка в списке заметок пользователя должна быть удалена. После удаления выделенной должна стать следующая строка. Если удаляется последняя не единственная в списке строка, выделенной должна стать строка, ставшая последней после удаления. Выделенная строка должна отобразиться в окне редактирования.
  4. Написать обработчик события нажатия на кнопку «Очистить»: содержимое текущей вкладки ежедневника удаляется.
  5. Модифицировать программу следующим образом: добавление записи из окна редактирования происходит при нажатии клавиши ввода.
  6. Написать обработчик события нажатия на кнопку «Закрыть»: приложение закрывается, данные текущего дня ежедневника автоматически сохраняются в соответствующий файл.
  7. Добавьте к приложению строку состояния (StatusStrip), в которой отобразите текущие дату, время и день недели.
  8. Напишите все необходимые обработчики исключительных ситуаций, чтобы приложение было устойчиво к ошибкам.

***Баллы:*** Приложение «Ежедневник» оценивается в 5 баллов