

Практическая работа №25

Разработка интерфейса оконных приложений

1 Цель работы

- 1.1 Научиться настраивать внешний вид приложений на языке C#;
- 1.2 Закрепить навык разработки оконных приложений на языке C#;
- 1.3 Закрепить навык разработки объектно-ориентированных программ на языке C#.

2 Литература

2.1 Фленов, М. Е. Библия C#. 3 изд. / М. Е. Фленов. – Санкт-Петербург: БХВ-Петербург, 2016. – Режим доступа: <https://ibooks.ru/reading.php?productid=353561>, только для зарегистрированных пользователей. – Загл. с экрана. – п.5.1-5.5, гл.11.

3 Задание

3.1 Создание оконного приложения и настройка формы стилей

3.1.1 Создать проект WinForms. Переименовать Form1 в MainForm (главную форму приложения).

3.1.2 В свойствах проекта указать значок для приложения (файл с расширением .ico, который будет использоваться как иконка исполняемого файла), предварительно скачать файл иконки из интернета.

3.1.3 Добавить в обозревателе решений в проект форму Windows с названием StyleForm (будет родительской для всех остальных форм, предназначена для настройки интерфейса приложения).

Настроить у StyleForm в дизайнере следующие свойства:

- StartPosition (положение при запуске): CenterScreen;
- Font (шрифт): Name (название шрифта) – на любой, поддерживающий русский язык и отличный от стандартного;
- BackColor (цвет фона): белый;
- MinimumSize (минимальный размер): ширина: 600, высота: 400;
- Icon (иконка формы): выбрать файл с расширением *.ico (предварительно скачать файл иконки из интернета, размер файла выбирать небольшой, например 32x32)

3.1.4 Добавить в форму StyleForm метод void SetStyle(Control control), в котором будут выполняться настройки элементов управления (здесь – меняется стиль всех кнопок формы).

Метод должен вызываться при загрузке формы:

SetStyle(this);

Написать в методе SetStyle следующий код для смены настроек кнопок:

```
foreach (Control item in control.Controls)
```

```
{
```

```
    if (item is Button)
```

```
    {
```

```
        var button = (Button)item;
```

```
        button.FlatStyle = FlatStyle.Flat;
```

```
        button.Height = 30;
```

```
        button.ForeColor = ColorTranslator.FromHtml("#132b4f"); // цвет в формате hex
```

```
        if (text.Contains("Отмена"))
```

```
        {
```

```

        button.BackColor = Color.Red;
    }
    else
    {
        button.BackColor = ColorTranslator.FromHtml("#f79420");// цвет в формате hex
    }
}
if (item.HasChildren)
{
    SetStyle(item);
}
}

```

3.2 Добавление элементов управления на форму стилей

3.2.1 На форму StyleForm добавить из вкладки Панель элементов элемент Panel (будет использоваться в качестве «шапки» формы), настроить у нее следующие свойства:

- BackColor (цвет фона): любой из вкладки Другой или Интернет;
- Size: Height (высота) – любое число от 50 до 150.
- Dock (размещение): Top (по верхнему краю).

3.2.2 Добавить на панель из вкладки Панель элементов элемент PictureBox (будет использоваться для отображения логотипа приложения) в левую часть панели, настроить у него следующие свойства:

- Image (изображение): выбрать растровый рисунок;
- SizeMode (режим подбора размера изображения): Zoom (масштабирование).

3.2.3 Добавить на панель из вкладки Панель элементов элемент Label, разместив его справа от PictureBox, настроить у него следующие свойства:

- Name (название): titleLabel;
- Text (подпись): Заголовок формы;
- Font (настройки шрифта): Size – подобрать такой, чтобы текст занимал примерно половину высоты панели, Bold (полужирный) – true;
- Modifiers (модификатор доступа): protected (чтобы элемент был доступен из дочерних форм).

3.2.4 Добавить на панель из вкладки Панель элементов элемент Button, разместив его в правой части панели, настроить у него следующие свойства:

- Text (подпись): Назад;
- Anchor (привязка): Top Right – элемент привязан к правому верхнему углу формы; Создать обработчик событий Click, в обработчике закрывать текущую форму.

3.2.5 На форму StyleForm добавить из вкладки Панель элементов элемент Panel (будет использоваться в качестве «подвала» формы), настроить у нее следующие свойства:

- BackColor (цвет фона): любой из вкладки Другой или Интернет;
- Size: Height (высота) – 50.
- Dock (размещение): Bottom (по нижнему краю).

3.2.6 Добавить на нижнюю панель элемент Label, настроить у него следующие свойства:

- AutoSize (подстрока размера под содержимое): False;
- Dock (размещение): Fill;
- TextAlign (выравнивание текста): MiddleCenter.

При загрузке формы в метке должна отображаться текущая дата.

3.3 Применение формы стилей и создание многостраничной формы

3.3.1 Изменить у MainForm родительскую форму на StyleForm и добавить на главную форму кнопки «Таблицы», «Авторизация» и «Регистрация».

3.3.2 Создать новую форму TablesForm и реализовать переход к ней с главной формы при нажатии на кнопку «Таблицы». Изменить у TablesForm родительскую форму на StyleForm.

3.3.3 Добавить на форму TablesForm TabControl и разместить в нем вкладки с подписями «Главная», «Товары», «Заказы», «Покупатели». Настроить у TabControl свойство Dock: Fill, чтобы элемент занимал всю форму.

3.3.4 На загрузку формы прописать код, скрывающий подписи к вкладкам:

```
компонент.Appearance = TabAppearance.FlatButtons;
```

```
компонент.ItemSize = new Size(0, 1);
```

```
компонент.SizeMode = TabSizeMode.Fixed;
```

3.3.5 Реализовать переход между вкладками:

- добавить на вкладку «Главная» три кнопки с подписями «Товары», «Заказы», «Покупатели» и реализовать переход к соответствующим вкладкам, изменяя свойство SelectedTab или SelectedIndex:

```
компонент.SelectedTab = tabPage2;
```

```
компонент.SelectedIndex = 2;
```

- добавить на все вкладки кроме вкладки «Главная» кнопку с подписью «Назад» и реализовать переход с текущей вкладки на главную вкладку при нажатии кнопки «Назад», изменяя свойство SelectedTab или SelectedIndex.

Для того, чтобы вкладки «Товары», «Заказы», «Покупатели» между собой отличались, на каждой изменить цвет фона вкладки.

3.3.6 При переходе между вкладками соответственно должен изменяться заголовок формы, отображаемый в свойстве Text формы и в titleLabel.

3.4 Создание диалоговых форм и переопределение методов родительской формы

3.4.1 Добавить в приложение форму DialogForm. Изменить у нее родительскую форму на StyleForm. Добавить в форму кнопки «ОК» и «Отмена», указать у них привязку к правому нижнему углу. У кнопки «ОК» указать в свойствах DialogResult = OK, у кнопки «Отмена» – Cancel.

Обработать событие FormClosing, чтобы запрашивалось подтверждение закрытия формы.

Добавить в DialogForm защищенный метод void ProcessData() и вызвать его при нажатии «ОК». Метод будет переопределяться в дочерних формах для обработки данных пользователя.

3.4.2 Добавить в приложение форму авторизации AuthorizationForm.

Реализовать открытие созданной формы из главной формы в диалоговом режиме при нажатии на кнопку «Авторизация». Изменить у AuthorizationForm родительскую форму на DialogForm. Заголовок формы требуется изменить на «Авторизация».

Настроить у формы свойство MaximumSize и MinimumSize (максимальный и минимальный размер): ширина: 500, высота: 400.

Создать обработчик загрузки формы и выполнить в обработчике изменение текста у элемента titleLabel на «Авторизация».

Добавить на форму два поля ввода с подписями в метках Логин и Пароль.

Переписать метод `ProcessData()`: в нем должно выводиться сообщение, что пользователь авторизован, если он заполнил поля логин и пароль, иначе — сообщение, что не удалось авторизоваться.

3.4.3 Добавить в приложение форму регистрации `RegistrationForm`. Настроить ее и реализовать ее открытие в диалоговом режиме из главной формы (аналогично описанию формы авторизации).

Переписать метод `ProcessData()`: в нем должно выводиться сообщение, что пользователь зарегистрирован, если он заполнил поля логин, пароль и подтверждение пароля, иначе — выдавать сообщение, что не удалось зарегистрироваться.

4 Порядок выполнения работы

4.1 Выполнить все задания из п.3 в одном решении `PractWork25`. Возможные ошибки требуется обрабатывать. Выполнить форматирование и рефакторинг кода.

4.2 Ответить на контрольные вопросы.

5 Содержание отчета

3.1 Титульный лист

3.2 Цель работы

3.3 Ответы на контрольные вопросы

3.4 Вывод

6 Контрольные вопросы

6.1 Какой класс является родительским для всех форм?

6.2 Какой метод содержит в себе описание настроек формы, сделанных в режиме дизайнера и где он вызывается по умолчанию?

6.3 Вызов каких методов выполняет открытие формы в диалоговом и недиалоговом режимах?

6.4 Вызов какого метода выполняет закрытие формы?

6.5 В каком обработчике событий формы можно отменить ее закрытие?

6.6 Как реализовать скрывание заголовков вкладок у `TabControl` в приложении `C#`?