# ЛАБОРАТОРНАЯ РАБОТА 17. ПРИМЕНЕНИЕ ЭЛЕМЕНТОВ УПРАВЛЕНИЯ В ПРИЛОЖЕНИЯХ WINDOWS

## 1. Цель и содержание

Цель лабораторной работы: изучить принципы использования стандартных элементов управления в приложениях Windows.

Задачи лабораторной работы:

– научиться использовать кнопки, фражки, переключатели;

– научиться использовать списки, выпадающие списки;

– научиться .

## 2. Формируемые компетенции

Лабораторная работа направлена на формирование следующих компетенций:

– способность к проектированию базовых и прикладных информационных технологий (ПК-11);

– способность разрабатывать средства реализации информационных технологий (методические, информационные, математические, алгоритмические, технические и программные) (ПК-12).

## 3. Теоретическая часть

Понимание иерархии классов элементов управления Windows имеет важное значение при проектировании интерфейсов пользователя, особенно при разработке собственных элементов управления.

В пространстве имен System.Windoiws .Forms есть один особенный класс, служащий базовым для почти каждого создаваемого элемента управления и формы. Это System.Windows.Forms.Control. Класс Control реализует базовую

функциональность создания и отображения всего, что видит пользователь. Класс Control унаследован от класса System.ComponentModel.Component. Класс Component обеспечивает Control всей необходимой инфраструктурой, которая потребуется для того, чтобы его можно было перетаскивать на поверхность проектирования в визуальном конструкторе и помещать в другой объект. Класс Control предоставляет огромный список функциональности классам, унаследованным от него.

Размер и местоположение элемента управления определяется свойствами Height, Width, Top, Bottom, Left и Right, наряду с дополняющими их свойствами Size и Location. Отличие между ними в том, что свойства Height, Width, Top, Bottom, Left и Right принимают целочисленные значения, Size – структуру Size, а Location – структуру Point. Структуры Srze и Point содержат версии координат X,Y. Структура Point обычно связана с местоположением, а Size задает высоту и широту объекта. Size и Point определены в пространстве имен System. Drawing. Обе структуры очень похожи в том отношении, что представляют пару координат X,Y, но также имеют перегруженные операции для облегчения сравнения и преобразования. Так, например, две структуры Size можно складывать вместе. В структуре Point операция Addition переопределена так, что можно прибавлять структуру Size к Point и получать новое значение Point. Это обеспечивает эффект прибавления расстояния к местоположению с получением нового местоположения, что очень удобно, если нужно динамически создавать формы и элементы управления.

Свойство Bounds возвращает объект Rectangle, представляющий область элемента управления. Эта область включает линейки прокрутки и панель заголовка. Rectangle – также часть пространства имен System. Drawing. Свойство ClientSize – это структура Size, представляющая клиентскую область элемента управления, минус линейки прокрутки и панель заголовка.

PointToClient и PointToScreen – методы преобразования, которые принимают Point и возвращают Point. Метод PointToClient принимает Point, представляющую экранные координаты, и транслирует их в координаты,

основанные на текущем клиентском объекте. Это удобно для операций перетаскивания. Метод PointToScreen выполняет прямо противоположную операцию – принимает координаты клиентского объекта и транслирует их в экранные координаты. Методы RectangleToScreen и ScreenToRectangle выполняют те же функции со структурами Rectangle вместо Point.

Свойство Anchor прикрепляет край элемента управления к краю родительского элемента. Это отличается от стыковки тем, что не устанавливает грань по родительскому элементу, а устанавливает некоторую постоянную дистанцию от края. Например, если после прикрепления правого края элемента управления к правому краю родительского элемента размер родительского элемента будет изменен, правый край дочернего элемента останется на том же расстоянии от правого края родительского элемента. Свойство Anchor принимает значение типа перечисления AnchorStyle. Перечисление включает следующие значения: Top, Bottom, Left, Right и None.

С внешним видом элемента управления связаны свойства BackColor и ForeColor, которые принимают значения типа System. Drawing.Color. Свойство BackGroundlmage принимает в качестве значения объект на базе Image. Класс System. Drawing. Image – это абстрактный класс, используемый в качестве базового для классов Bitmap и Metafile.

Свойство BackgroundlmageLayout использует перечисление ImageLayout для установки режима вывода графического изображения в элементе управления. Допустимыми значениями являются Center,Tile, Stretch, Zoom и None.

Свойства Font и Text имеют дело с отображением текста. Чтобы изменить Font, потребуется создать объект Font. При создании объекта Font указывается имя шрифта, размер и стиль.

Для выполнения лабораторной работы необходимо изучить теоретические основы проектирования с использованием стандартных элементов управления библиотеки .NET Framework.

Класс. Button. Класс Button представляет простую командную кнопку и наследуется от класса ButtonBase. Наиболее часто приходится писать код для обработки события Click кнопки. В следующем фрагменте кода реализован обработчик события Click (обработка данного события не представляет сложностей). С помощью метода PerformClick можно эмулировать событие Click на кнопке без реального щелчка на ней пользователем, что может пригодиться для тестирования построенного пользовательского интерфейса. В окне также имеется понятие кнопки по умолчанию, на которой автоматически совершается щелчок, если пользователь нажимает в этом окне клавишу «Enter». Чтобы идентифицировать кнопку как кнопку по умолчанию, в форме, содержащей эту кнопку, необходимо установить свойство AcceptButton в объект кнопки.

Класс CheckBox (рис. 22.1). Элемент управления CheckBox также унаследован от ButtonBase и используется для приема двух или трех состояний от пользователя.

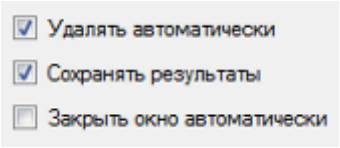



Рисунок 22.1 – Внешний вид CheckBox.

Если установить свойство ThreeState в true, то свойство CheckState элемента CheckBox сможет принимать одно их трех значений перечисления CheckState, описанных в табл. 22.1.

Таблица 22.1 – Состоятия CheckBox (возможные значения перечисления CheckState).

| Значение | Описание |
|---|---|
| Checked | Флажок имеет отметку. |
| Unchecked | Флажок не имеет отметки. |
| Indeterminate | В этом состоянии флажок становится серым (неопределенное состояние). |

Состояние Indeterminate удобно, когда пользователь должен быть уведомлен, что опция не установлена. Если нужно булевское значение, можно проверить свойство Checked.

События CheckedChanged и CheckStateChanged возникают, когда изменяются значения свойств CheckState и Checked. Для флажка с тремя состояниями понадобится присоединиться к событию CheckStateChanged. Перехват этих событий может быть полезен для установки других значений, основанных на новом состоянии CheckBox.

Класс RadioButton (рис. 22.2). Переключатель (кнопки опций) – элемент управления, унаследованный от ButtonBase.

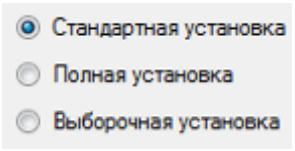



Рисунок 22.2 – Внешний вид RadioButton.

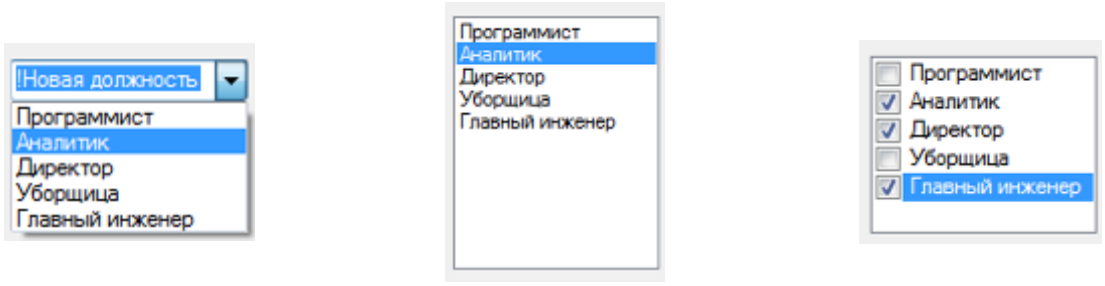
Переключатели обычно используются в группе. Переключатели позволяют пользователю выбирать одну из нескольких опций. При наличии нескольких элементов управления RadioButton в одном контейнере, только один из них может быть выбран в один и тот же момент времени.

Свойство Appearance принимает значение перечисления Appearance, которым может быть Button либо Normal. В случае установки значения Normal переключатель выглядит как маленький кружочек с меткой рядом с ним. Выбор переключателя заполняет этот кружок, а выбор другого переключателя снимает отметку с текущего выбранного переключателя и кружок делается пустым. В случае установки значения Button элемент управления выглядит подобно стандартной кнопке, но работает как переключатель – его выбор означает включение (вдавливание кнопки), а отказ от выбора – выключение (стандартное положение кнопки).

Свойство CheckedAlign определяет местоположение кружка по отношению к тексту метки. Он может быть над меткой, с любой стороны от нее либо под меткой.

Событие CheckedChanged инициируется при любом изменении свойства Checked. Это позволяет предпринимать другие действия на основе нового значения элемента управления.

Классы ComboBox, ListBox и CheckedListBox (рис. 22.3). Элементы управления ComboBox, ListBox и CheckedListBox унаследованы от класса ListControl. Этот класс предоставляет некоторую базовую функциональность управления списками. Наиболее важные аспекты использования списочных элементов управления состоят в добавлении данных и выборе данных из списка.




а) б) в)

Рисунок 22.3 – Внешний вид списков: а – ComboBox; б – ListBox; в – CheckedListBox.

На выбор списка влияет то, как он должен использоваться, и тип данных, которые будут помещены в список. Если есть необходимость во множественном выборе, или если пользователю нужно видеть несколько элементов в списке в одно и то же время, то для этого лучше всего подойдут ListBox и CheckedListBox. Если в любой момент времени в списке может быть выбран только один элемент, то предпочесть следует ComboBox.

Для добавления элементов в списки необходимо обратиться к коллекции ListBox.ObjectCollection, которая доступна через свойство Items списка.

Поскольку коллекция хранит объекты, в список может быть добавлен любой допустимый тип .NET. Чтобы идентифицировать элементы, необходимо установить два важных свойства. Первым свойством является DisplayMember. Его настройка сообщает ListControl, какое свойство объекта должно быть отображено в списке. Другое свойство – ValueMember. Оно представляет собой свойство объекта, которое должно возвращаться в качестве значения. Если в список были добавлены строки, для обоих свойств по умолчанию используются строковые значения.

После загрузки данных в список для их получения могут быть использованы свойства Selectedltem и Selectedlndex. Свойство Selectedltem возвращает объект, выбранный в данный момент. Если список предполагает множественный выбор, то должна применяться коллекция Selectedltems для получения выбранных элементов.

Если для наполнения списка применяется DataBinding, то свойство SelectedValue вернет значение свойства выбранного объекта, которое было установлено в свойстве ValueMember.

Элемент ComboBox – это комбинация поля редактирования и окна списка. Стиль ComboBox задается установкой в свойстве DropDownStyle значения перечисления DropDownStyle (таблица 22.2).

Таблица 22.2 – Стили ComboBox (возможные значения перечисления DropDownStyle).

| Значение | Описание |
|---|---|
| DropDown | Текстовая часть комбинированного списка допускает редактирование, и пользователи могут вводить значение. Также они могут щелкать на кнопке со стрелочкой, чтобы отобразить раскрывающийся список элементов. |
| DropDownList | Текстовая часть не допускает редактирование. Пользователи должны выбирать значения из списка. |
| Simple | Подобно DropDown, но список является постоянно видимым. |

Класс DateTimePicker (рис. 22.4). Элемент управления DateTimePicker позволяет выбирать значение даты или времени (или то и другое) во множестве разных форматов. Значение DateTime можно отобразить в любом стандартном формате времени и даты. Свойство Format принимает значение перечисления DateTimePickerFormat, которое устанавливает формат Long, Short, Time или Custom. Если свойство Format установлено в DateTimePickerFormat .Custom, с помощью свойства CustomFormat можно задать строку, представляющую формат.

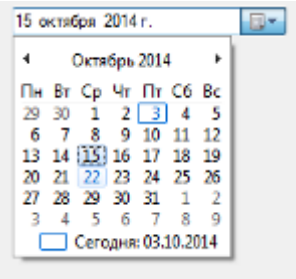



Рисунок 22.4 – Внешний вид DateTimePicker.

Свойство Text возвращает строковое представление значения DateTime, а свойство Value – объект DateTime. С помощью свойств MinDate и MaxDate можно устанавливать максимальное и минимальное значения дат. Когда пользователь щелкает на стрелке вниз, отображается календарь, позволяющий выбрать дату. Доступны свойства, которые позволяют изменить внешний вид календаря, указать фоновые цвета заголовка и месяца, а также цвета переднего плана.

## 4. Оборудование и материалы

Для выполнения лабораторной работы рекомендуется использовать персональный компьютер со следующими характеристиками: 64-разрядный (x64) процессор с тактовой частотой 1 ГГц и выше, оперативная память – 1 Гб и выше, свободное дисковое пространство – не менее 1 Гб, графическое

устройство DirectX 9. Программное обеспечение: операционная система WINDOWS 7 и выше, Microsoft Visual Studio 20112 и выше.

## 5. Указания по технике безопасности

Техника безопасности при выполнении лабораторной работы определяется общепринятой для пользователей персональных компьютеров. Самостоятельно не производить ремонт персонального компьютера, установку и удаление программного обеспечения; в случае неисправности персонального компьютера сообщить об этом обслуживающему персоналу лаборатории; не касаться электрических розеток металлическими предметами; рабочее место пользователя персонального компьютера должно содержаться в чистоте; не разрешается возле персонального компьютера принимать пищу, напитки.

## 6. Методика и порядок выполнения работы

Для выполнения лабораторной работы необходимо выполнить следующую последовательность действий:

1. Создайте проект приложения Windows Forms в среде MS Visual Studio.

2. Реализуйте методы для вычисления выражений в соответствии с вариантом индивидуального задания.

3. При проектировании формы необходимо учитывать следующие особенности задачи:

– выражение может быть вычислено двумя различными способами, то есть пользователь должен иметь возможность выбрать метод расчета (подумайте, какой элемент управления подходит для этого больше всего?);

– в правой части выражения присутствуют неизвестные переменные, которые пользователь может вводить с клавиатуры в текстовые поля;

– в некоторых выражениях присутствуют коэффициенты, выбор значений которых необходимо реализовать с помощью различных списков.

**Индивидуальное задание.**

| Вариант | Выражение для вычисления |
|---|---|
| 1 | $Z=\begin{cases}1-\frac{X}{2}+\frac{Y^2}{6}-\frac{X^3}{24}+\frac{Y^4}{120}-..;. \\ \sum_{i=1}^{N}\sum_{j=1}^{R}\frac{3+a\cdot j}{b\cdot i}.\end{cases}$ |
| 2 | $Z=\begin{cases}-\frac{1}{2}+\frac{Y}{6}-\frac{X^2}{24}+\frac{Y^3}{120}-..;. \\ \sum_{i=1}^{N}\sum_{j=1}^{R}\frac{i^2+j}{i^j}.\end{cases}$ |
| 3 | $Z=\begin{cases}-\frac{1}{1}+\frac{Y}{2}-\frac{X^2}{3}+\frac{Y^3}{4}-..;. \\ \sum_{i=1}^{N}\sum_{j=1}^{R}\frac{j^2+h\cdot i}{i^j+c\cdot j}.\end{cases}$ |
| 4 | $Z=\begin{cases}-\frac{Y}{2}+\frac{Y^2}{6}-\frac{X^3}{24}+\frac{Y^4}{120}-..;. \\ \sum_{i=1}^{N}\sum_{j=1}^{R}\frac{a\cdot i+b\cdot j}{c\cdot i^j}.\end{cases}$ |
| 5 | $Z=\begin{cases}-\frac{p^2}{2}+\frac{p^3}{6}-\frac{p^4}{24}+\frac{p^5}{120}-..;. \\ \sum_{i=1}^{N}\sum_{j=1}^{R}\frac{i^2+b\cdot j}{c\cdot i^3}.\end{cases}$ |
| 6 | $Z=\begin{cases}-\frac{X}{2}+\frac{p\cdot X^2}{6}-\frac{p^2\cdot X^3}{24}+\frac{p^3\cdot X^4}{120}-..;. \\ \sum_{i=1}^{N}\sum_{j=1}^{R}\frac{i^2+a\cdot j^2}{i^3+j^3}.\end{cases}$ |

| | |
|---|---|
| 7 | $Z=\begin{cases}-\frac{X}{2}+\frac{X^2}{6}-\frac{X^3}{24}+\frac{X^4}{120}-..;. \\ \sum_{i=1}^{N}\sum_{j=1}^{R}\frac{a\cdot i^2}{i^3+b\cdot j^3}.\end{cases}$ |
| 8 | $Z=\begin{cases}-\frac{1}{2}+\frac{X}{6}-\frac{X^2}{24}+\frac{X^3}{120}-..;. \\ \sum_{i=1}^{N}\sum_{j=1}^{R}\frac{a\cdot i}{i^3+j^3}.\end{cases}$ |
| 9 | $Z=\begin{cases}-\frac{f}{2}+\frac{X\cdot f^2}{6}-\frac{X^2\cdot f^3}{24}+\frac{X^3\cdot f^4}{120}-..;. \\ \sum_{i=1}^{N}\sum_{j=1}^{R}\frac{a\cdot(i+j)}{i\cdot(i+2)}.\end{cases}$ |
| 10 | $Z=\begin{cases}1-\frac{X}{2}+\frac{Y^2}{6}-\frac{X^3}{24}+\frac{Y^4}{120}-..;. \\ \sum_{i=1}^{N}\sum_{j=1}^{R}\frac{3+j}{c\cdot i}.\end{cases}$ |
| 11 | $Z=\begin{cases}-\frac{1}{2}+\frac{Y}{6}-\frac{X^2}{24}+\frac{Y^3}{120}-..;. \\ \sum_{i=1}^{N}\sum_{j=1}^{R}\frac{i^2+j}{a^j}.\end{cases}$ |
| 12 | $Z=\begin{cases}-\frac{1}{1}+\frac{Y}{2}-\frac{X^2}{3}+\frac{Y^3}{4}-..;. \\ \sum_{i=1}^{N}\sum_{j=1}^{R}\frac{j^2+c\cdot i}{i^j+d\cdot j}.\end{cases}$ |
| 13 | $Z=\begin{cases}-\frac{Y}{2}+\frac{Y^2}{6}-\frac{X^3}{24}+\frac{Y^4}{120}-..;. \\ \sum_{i=1}^{N}\sum_{j=1}^{R}\frac{a\cdot i+b\cdot j}{c\cdot i^j}.\end{cases}$ |
| 14 | $Z=\begin{cases}-\frac{p^2}{2}+\frac{p^3}{6}-\frac{p^4}{24}+\frac{p^5}{120}-..;. \\ \sum_{i=1}^{N}\sum_{j=1}^{R}\frac{i^2+b\cdot j}{c^i\cdot i^3}.\end{cases}$ |

| | |
|---|---|
| 15 | $Z=\begin{cases}-\frac{X}{2}+\frac{p\cdot X^2}{6}-\frac{p^2\cdot X^3}{24}+\frac{p^3\cdot X^4}{120}-..;.\\ \sum_{i=1}^{N}\sum_{j=1}^{R}\frac{i^2+j^2}{i^3+j^3}.\end{cases}$ |
| 16 | $Z=\begin{cases}-\frac{X}{2}+\frac{2\cdot X^2}{6}-\frac{3\cdot X^3}{24}+\frac{4\cdot X^4}{120}-..;.\\ \sum_{i=1}^{N}\sum_{j=1}^{R}\frac{(i+1)^2}{a\cdot i^3+b\cdot j^3}.\end{cases}$ |
| 17 | $Z=\begin{cases}-\frac{1}{2}+\frac{X}{6}-\frac{X^2}{24}+\frac{X^3}{120}-..;.\\ \sum_{i=1}^{N}\sum_{j=1}^{R}\frac{a\cdot i}{i^3+b\cdot j^3}.\end{cases}$ |
| 18 | $Z=\begin{cases}-\frac{f}{2}+\frac{X\cdot f^2}{6}-\frac{X^2\cdot f^3}{24}+\frac{X^3\cdot f^4}{120}-..;.\\ \sum_{i=1}^{N}\sum_{j=1}^{R}\frac{a\cdot(i+j)}{b\cdot i\cdot(i+2)}.\end{cases}$ |
| 19 | $Z=\begin{cases}-\frac{1}{1}+\frac{Y}{3}-\frac{X^2}{5}+\frac{Y^3}{7}-..;.\\ \sum_{i=1}^{N}\sum_{j=1}^{R}\frac{a\cdot j^2+2}{b\cdot i^j+3}.\end{cases}$ |
| 20 | $Z=\begin{cases}-\frac{Y}{2}+\frac{Y^2}{6}-\frac{X^3}{24}+\frac{Y^4}{120}-..;.\\ \sum_{i=1}^{N}\sum_{j=1}^{R}\frac{a\cdot i+b\cdot j}{c\cdot i^j}.\end{cases}$ |
| 21 | $Z=\begin{cases}-\frac{p^2}{1\cdot 2}+\frac{p^3}{2\cdot 6}-\frac{p^4}{3\cdot 24}+\frac{p^5}{4\cdot 120}-..;.\\ \sum_{i=1}^{N}\sum_{j=1}^{R}\frac{i^2+d\cdot j}{c^i\cdot i^3}.\end{cases}$ |
| 22 | $Z=\begin{cases}-\frac{X}{2}+\frac{p\cdot X^2}{6}-\frac{p^2\cdot X^3}{24}+\frac{p^3\cdot X^4}{120}-..;.\\ \sum_{i=1}^{N}\sum_{j=1}^{R}\frac{i^2+j^2}{i^3}.\end{cases}$ |

| | |
|---|---|
| 23 | $Z=\begin{cases}-\frac{3\cdot Y}{2}+\frac{5\cdot Y^2}{6}-\frac{7\cdot X^3}{24}+\frac{9\cdot Y^4}{120}-..;.\\ \sum_{i=1}^{N}\sum_{j=1}^{R}\frac{i+k\cdot j}{c\cdot i^j}.\end{cases}$ |
| 24 | $Z=\begin{cases}-\frac{3\cdot p^2}{2}+\frac{5\cdot p^3}{6}-\frac{7\cdot p^4}{24}+\frac{9\cdot p^5}{120}-..;.\\ \sum_{i=1}^{N}\sum_{j=1}^{R}\frac{i^2+b}{c^i\cdot(i+1)^3}.\end{cases}$ |
| 25 | $Z=\begin{cases}-\frac{X}{2}+\frac{p\cdot X^2}{6}-\frac{p^2\cdot X^3}{24}+\frac{p^3\cdot X^4}{120}-..;.\\ \sum_{i=1}^{N}\sum_{j=1}^{R}\frac{(a\cdot i+j)^2}{i^3+b\cdot j^3}.\end{cases}$ |

7. Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы.

2. Цели лабораторной работы.

3. Ответы на контрольные вопросы.

4. Экранные формы и листинг программного кода, показывающие порядок выполнения лабораторной работы, и результаты, полученные в ходе её выполнения.

Отчет о выполнении лабораторной работы в письменном виде сдается преподавателю.

8. Контрольные вопросы

1. Какое событие элемента управления Button обрабатывается в программах чаще всего?

2. Для чего предназначен компонент CheckBox? Назовите основные свойства класса CheckBox.

3. Опишите назначение элемента RadioButton. Какой внешний вид может принимать данный компонент? Назовите основные свойства класса RadioButton.

4. Назовите классы компонентов для представления списочной информации.

5. Опишите основные свойства класса ListBox. Чем комопнет ListBox отличается от CheckedListBox?

## 9. Список литературы

Для выполнения лабораторной работы, при подготовке к защите, а также для ответа на контрольные вопросы рекомендуется использовать следущие источники: [6-8].