

Проектирование интерфейса  
оконного приложения  
с использованием элементов  
управления

**11 класс**

## Сегодня на уроке мы...

- ознакомимся с основными элементами управления: кнопка (Button), метка (Label), текстовое поле (Edit);
- рассмотрим примеры создания проектов с элементами управления;
- научимся проектировать интерфейс оконного приложения с использованием элементов управления.



# **Основные элементы управления**

**Элементами управления** называются объекты, которые используются для отображения данных или организации взаимодействия между пользователем и приложением с помощью мыши или клавиатуры.

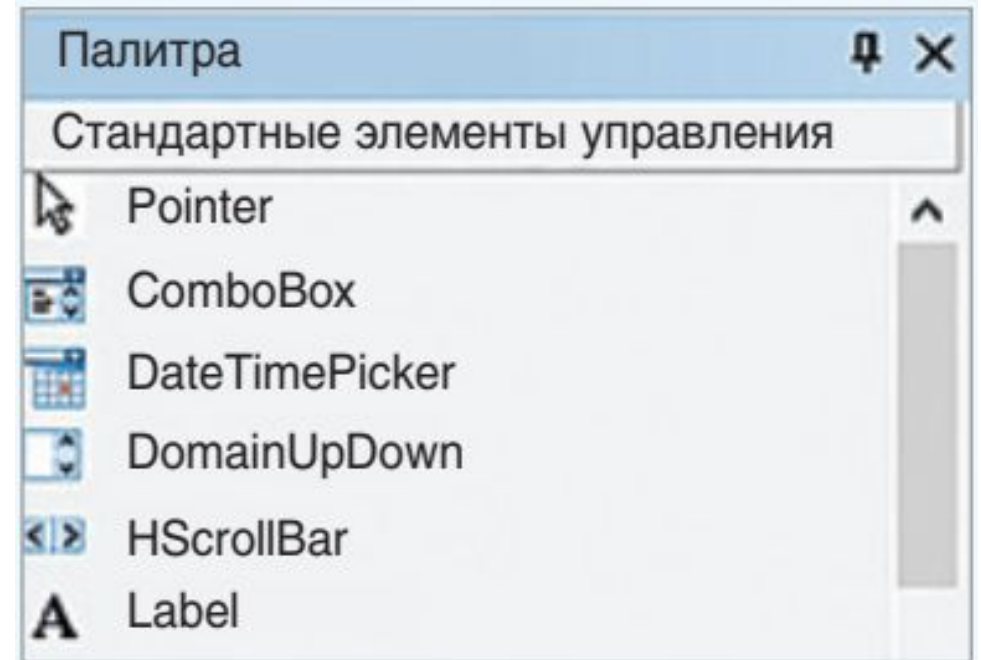
Элементами управления собраны в специальные библиотеки компонентов, которые ОС использует для обеспечения единообразного интерфейса прикладных программ. Наиболее распространенными элементами управления являются: *кнопки, редактируемые поля, списки выбора, флажки, переключатели* и т. д.

Компоненты библиотеки размещаются на различных страницах палитры компонентов. Каждая страница имеет свое название.

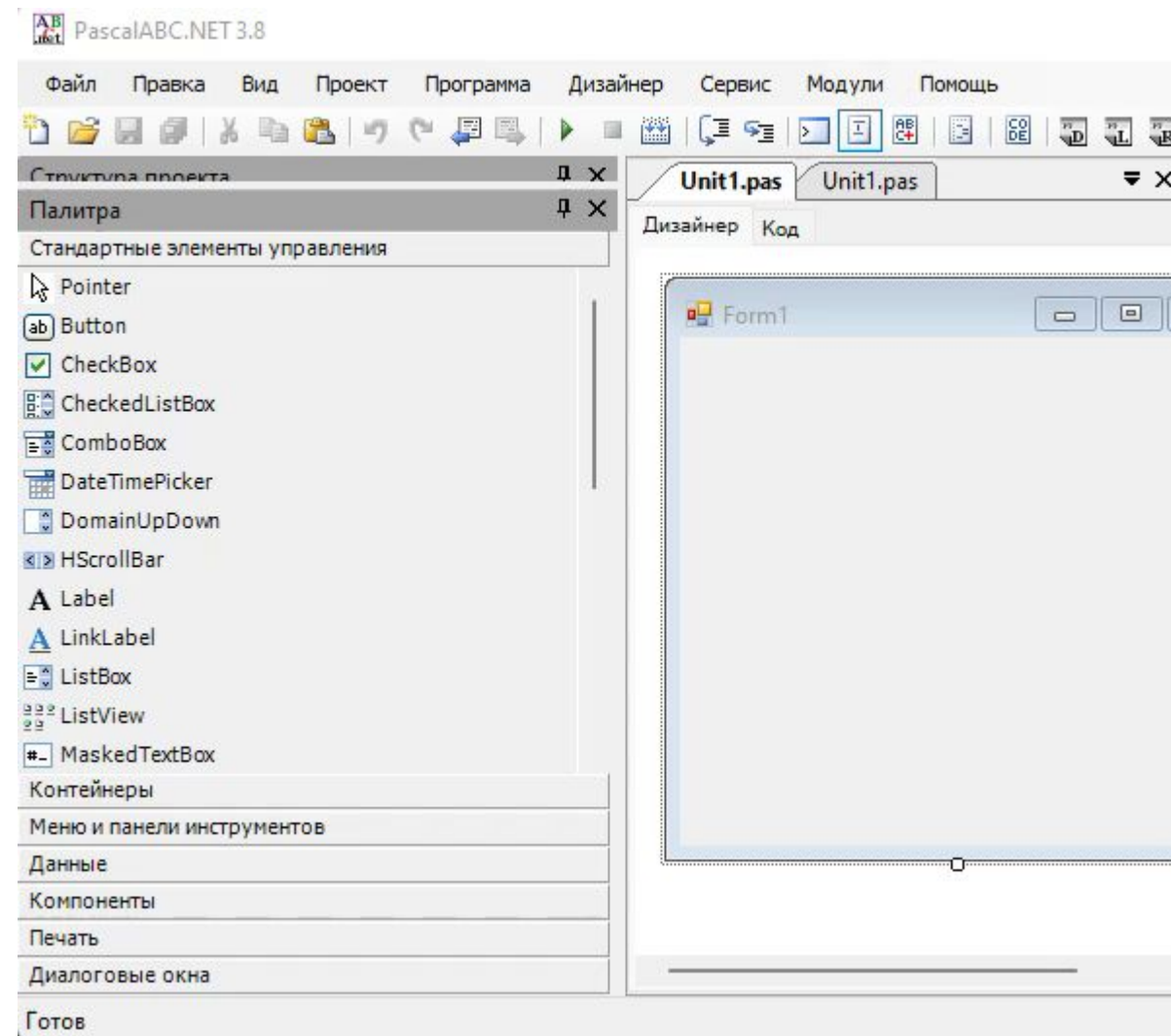
Для элементов управления используется и другое название — **виджеты**. Слово употребляется примерно с 1920-х гг. в американском английском для обозначения простой, но необходимой вещи, маленького изделия. Одним из вариантов происхождения этого слова считается словослияние «**window gadget**» (букв. «*оконное приспособление*»), также произошедшее в начале XX в.

*Стандартные элементы управления размещены наиболее употребляемые компоненты:*

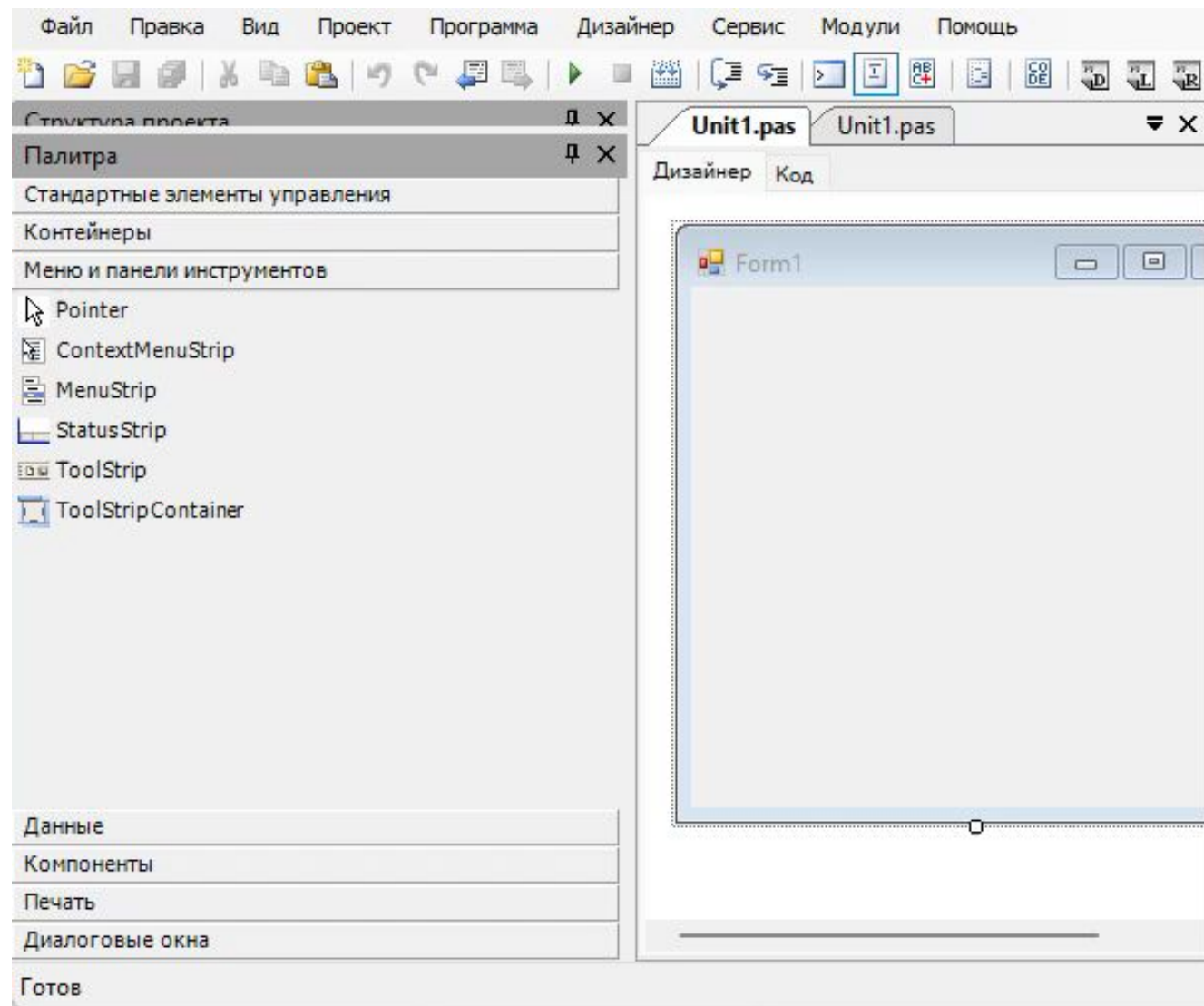
- **Кнопка** (Button)
- **Надпись** (Label)
- **Поле для ввода текста** (TextBox)
- **Флажок** (CheckBox)
- **Радиокнопка** (RadioButton)
- **Список** (ListBox)
- **Выпадающий список** (ComboBox)
- **Рисунок** (PictureBox)



Одним из наиболее используемых компонентов палитры **Компоненты** является компонент **Таймер** (*Timer*). Палитра **Меню и панели инструментов** содержит компоненты, необходимые для создания главного меню программы или контекстных меню для различных объектов, помещенных на форму.



Палитры **Печать** и **Диалоговые окна** содержат компоненты, обеспечивающие стандартные диалоги операционной системы: открытие и сохранение файла, выбор цвета, установки параметров шрифта, настройки принтера и управление печатью. Палитра **Данные** содержит компоненты для работы с таблицами баз данных.

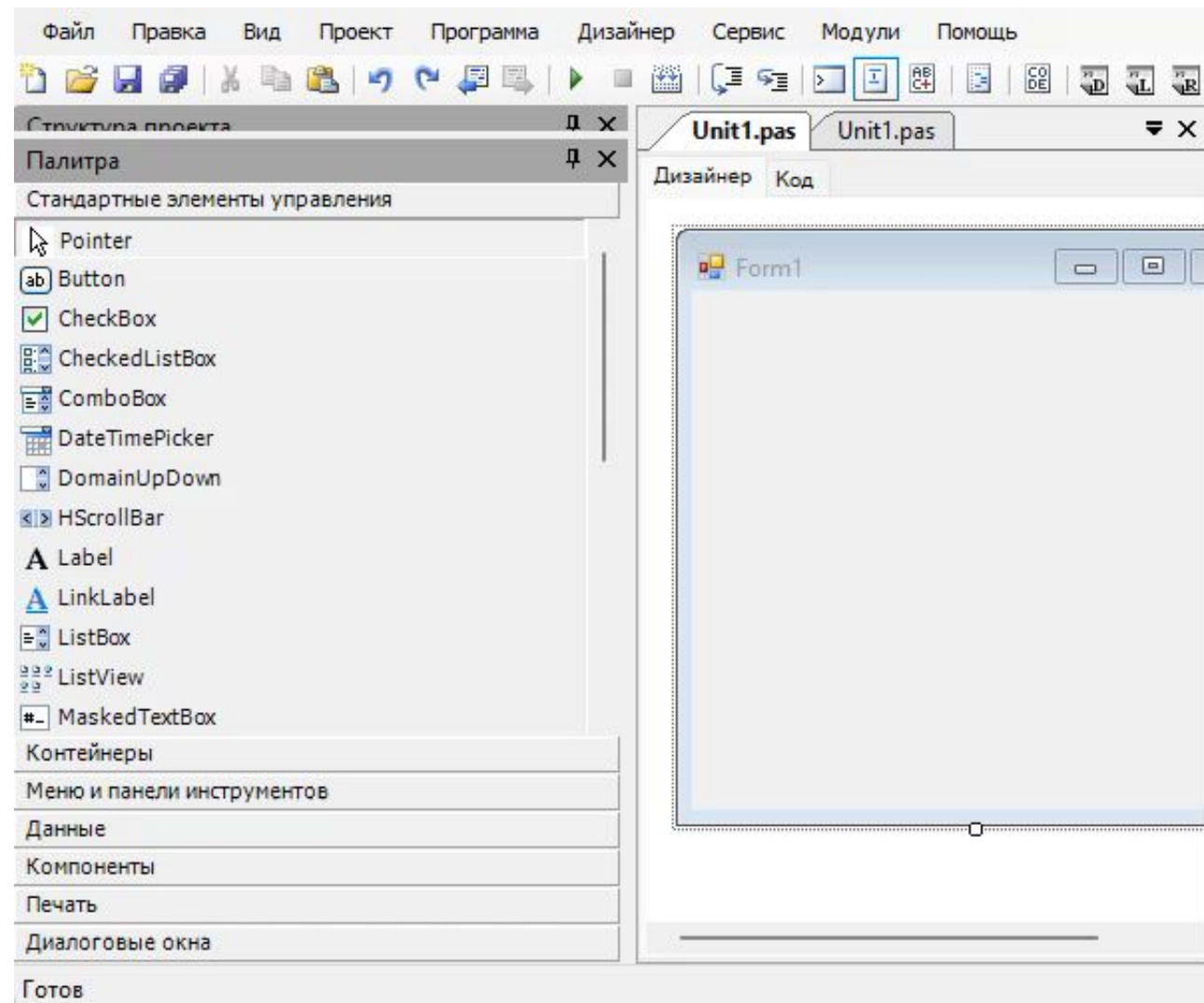




# Элемент управления кнопка (**Button**)

Компонент **кнопка** относится к элементам управления. На панели компонентов **Стандартные элементы** управления кнопка изображена в виде ab Button, имя объекта — *button*.

Кнопка, помещенная на форму, получает имя *buttonN*, где *N* — номер 1, 2, 3... При необходимости кнопку можно переместить в любое место формы. Ключевые точки позволят установить нужный размер кнопки.




Некоторые свойства кнопки перечислены в таблице:

*Свойства кнопки*

Некоторые свойства кнопки Button	
Свойство	Назначение
Text	Заголовок (внешнее имя) кнопки, текст, который отображается на кнопке. По умолчанию — button1
Font	Свойства шрифта для подписи заголовка. Свойство Font является сложным, о чем свидетельствует значок  , при нажатии на который раскрываются все свойства шрифта
Height	Высота кнопки
Weight	Ширина кнопки

Left	Горизонтальная координата положения верхнего левого угла кнопки на форме
Top	Вертикальная координата положения верхнего левого угла кнопки на форме
(Name)	Имя (внутреннее) кнопки. Используется в программном коде для обращения к объекту. Является идентификатором
Enabled	Значение True этого свойства обеспечивает доступность кнопки для мыши или клавиатуры
Visible	Значение True этого свойства обеспечивает видимость кнопки во время выполнения приложения

Как видно из таблицы, многие свойства кнопки совпадают по именам и назначениям со свойствами формы, поэтому в дальнейшем для компонентов будут указываться только те свойства, которые отличны от уже описанных для других компонентов.

Основным событием кнопки является *Click*. Для создания обработчика события *Click* для кнопки можно поступить так же, как и при создании аналогичного обработчика для формы: выбрать событие на вкладке **События** () и выполнить двойной щелчок в поле напротив события *Click*.

Можно просто выполнить двойной щелчок по кнопке. (Для формы основным событием является событие *Load*, поэтому при двойном щелчке по форме создается обработчик события *Load*.)

## Пример 1

Создать проект, разместив на форме две кнопки. При нажатии на одну из них цвет формы должен измениться на синий, а при нажатии на вторую — должен восстановиться исходный цвет.

### *Этапы выполнения задания*

1. Создать на форме две кнопки.
2. Изменить свойство *Text* у кнопки *button1* на **Изменить цвет**.
3. Изменить свойство *Text* у кнопки *button2* на **Восстановить цвет**.
4. Создать обработчик события *Click* для кнопки *button1* и изменить цвет формы. Команда ***BackColor := Color.Blue;***
5. Создать обработчик события *Click* для кнопки *button2* и изменить цвет формы на первоначальный (*название цвета формы указано в поле Color инспектора объектов*).

Команда ***BackColor := SystemColors.Control;***

# Пример 1

Создать проект, разместив на форме две кнопки. При нажатии на одну из них цвет формы должен измениться на синий, а при нажатии на вторую — должен восстановиться исходный цвет.

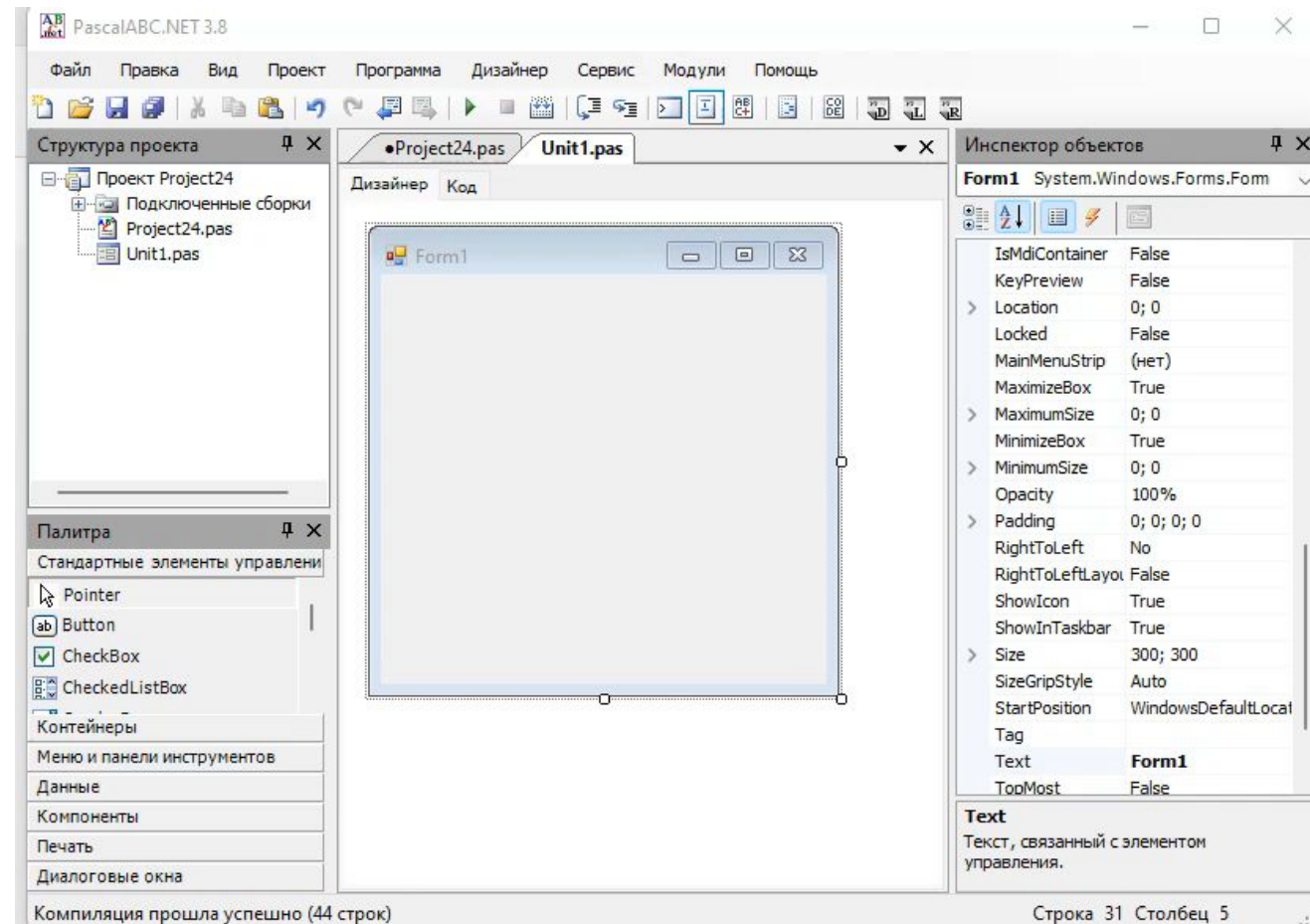
## Этапы выполнения задания

6. Сохранить изменения в проекте.

Название цвета ***SystemColors***.

***Control*** задает не какой-то определенный цвет. Это цвет элемента управления, заданный цветовой схемой Windows.

Поэтому он не обязательно будет серым.



# Пример 1


## Свойство кнопки

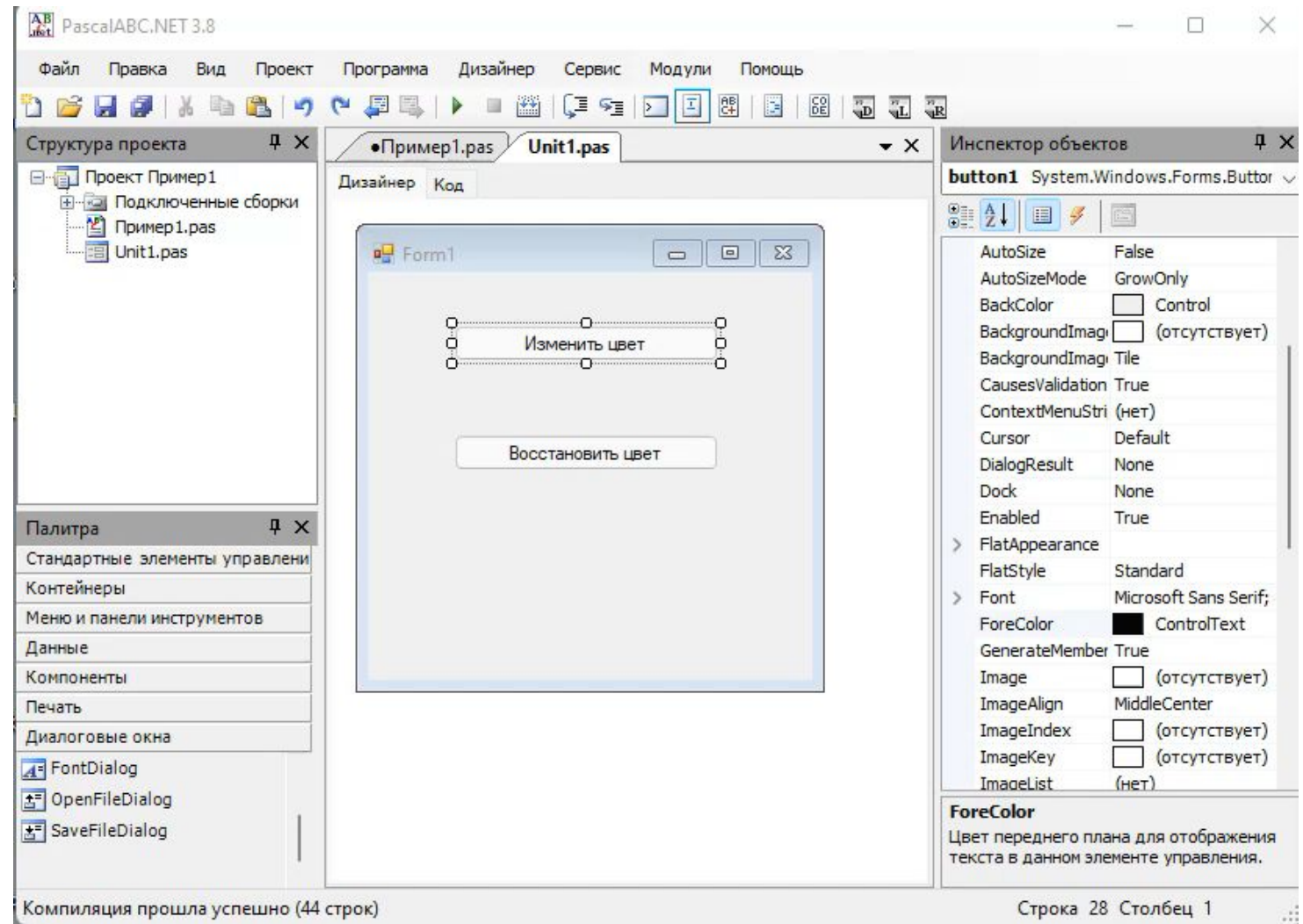
*BackgroundImage* позволяет


вставить на кнопку

изображение, хранящееся в графическом файле. Для

вставки можно использовать

кнопку . Далее выбрать файл с рисунком.



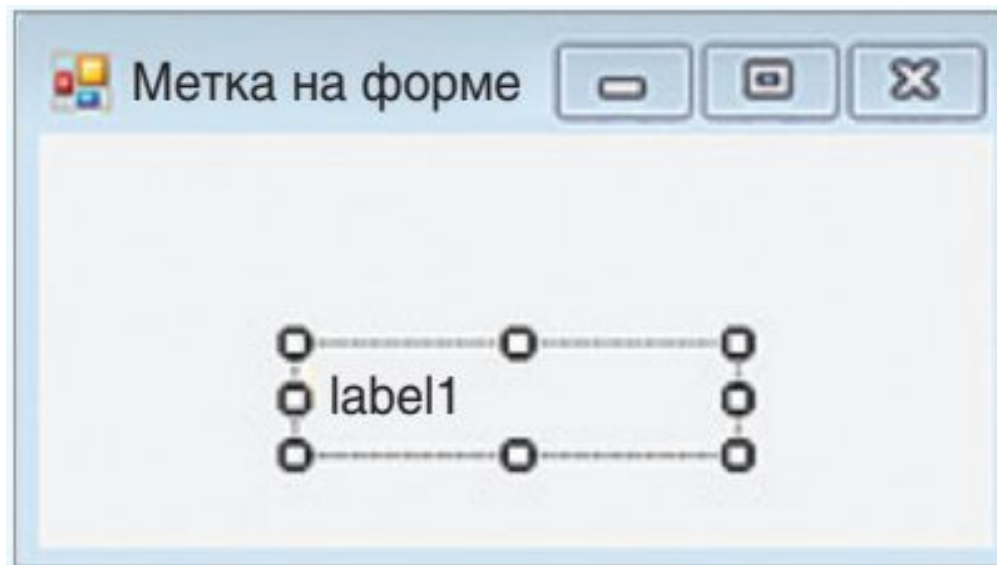


# Элемент управления метка (**Label**)



Компонент **метка** предназначен для отображения текста на форме. На панели компонентов **Стандартные элементы управления** метка изображена в виде , имя объекта — *label*.

Кнопка, помещенная на форму, получает имя *labelN*, где *N* — номер *1, 2, 3*




Некоторые свойства метки, отличные от свойств кнопки, перечислены в таблице. Основным событием для метки является *Click*.

Свойство	Назначение
Text	Отображает введенный текст на форме
BackColor	Устанавливает цвет фона метки, который по умолчанию совпадает с цветом формы. Фон метки можно сделать прозрачным, задав свойству BackColor значение Color.Transparent
AutoSize	Значение true этого свойства приводит к автоматическому изменению размеров метки в соответствии с длиной текста
TextAlign	Выравнивание текста относительно границ метки: 

## Пример 2

Создать проект, в котором описана возможность выполнять следующие действия: после запуска программы в окне с именем «Работаем с кнопкой и меткой» при щелчке мыши по кнопке «Приветствие» появляется сообщение «Здравствуй, мир!».

### *Этапы выполнения задания*

1. Изменить свойство *Text* формы на «**Работаем с кнопкой и меткой**».
2. Добавить на форму кнопку *button1*.
3. Изменить свойство *Text* кнопки на «**Приветствие**».
4. Добавить на форму метку *label1*.
5. Изменить свойства шрифта для компонента *label1*. Нажать кнопку  в поле *Font* (цвет шрифта — синий, размер — 20, стиль — жирный курсив).
6. Очистить поле *Text* у метки.

## Пример 2

Создать проект, в котором описана возможность выполнять следующие действия: после запуска программы в окне с именем «Работаем с кнопкой и меткой» при щелчке мыши по кнопке «Приветствие» появляется сообщение «Здравствуй, мир!».

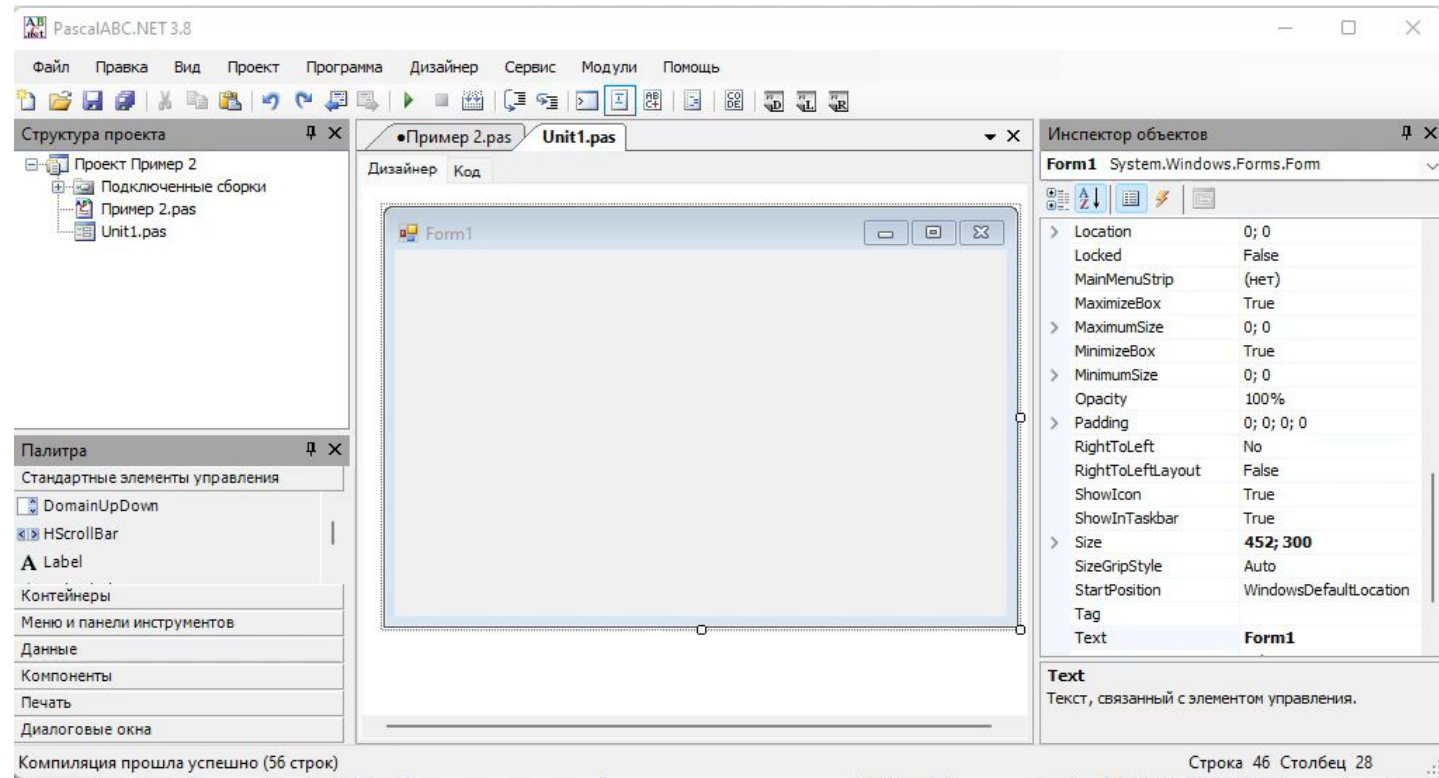
### Этапы выполнения задания

7. Установить значение *true* у свойства метки *Autosize*.

8. В обработчик события *Click* для кнопки *button1*

вписать команду

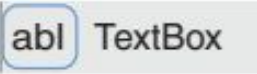
```
label1.Text := 'Здравствуй, мир!';
```

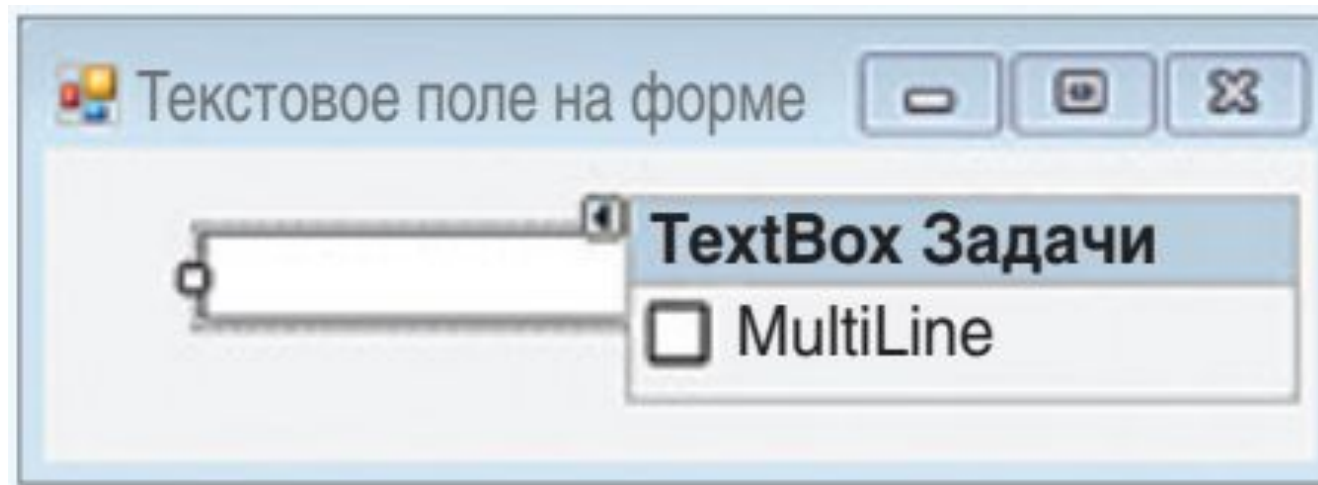




# Элемент управления текстовое поле (**Edit**)

**Текстовое поле** — компонент, который предназначен для ввода и вывода текстовой информации.

На панели компонентов **Стандартные элементы управления** текстовое поле изображено в виде , имя объекта — *TextBox*. Текстовое поле, помещенное на форму, получает имя *TextBoxN*, где *N* — номер 1, 2, 3...



В отличие от ранее рассмотренных компонентов, свойство *Text* у текстового поля по умолчанию пусто (у других — совпадает с именем компонента). Некоторые свойства компонента *TextBox* приведены в таблице.

Свойство	Назначение
BorderStyle	Определяет границу вокруг текстового поля: None (нет границы), FixedSingle или Fixed3d (есть граница)
MaxLength	Ограничивает количество символов, которые можно ввести в TextBox
ReadOnly	Значение true запрещает редактирование текста, отображаемого в TextBox
Text	Содержит вводимый или выводимый текст



Значение свойства *Text* компонента **текстовое поле** может изменяться программно или при вводе с клавиатуры. Основным событием для *TextBox* является *TextChanged*, которое происходит при изменении компонента. Наиболее часто программируют событие *KeyPress*, которое позволяет определить, какая клавиша была нажата.





## Пример 3

Создать проект, в котором пользователя попросят ввести его имя, а затем, после нажатия клавиши *Enter*, будет выдано сообщение «**Имя, приятно с Вами познакомиться!**»

### *Этапы выполнения задания*

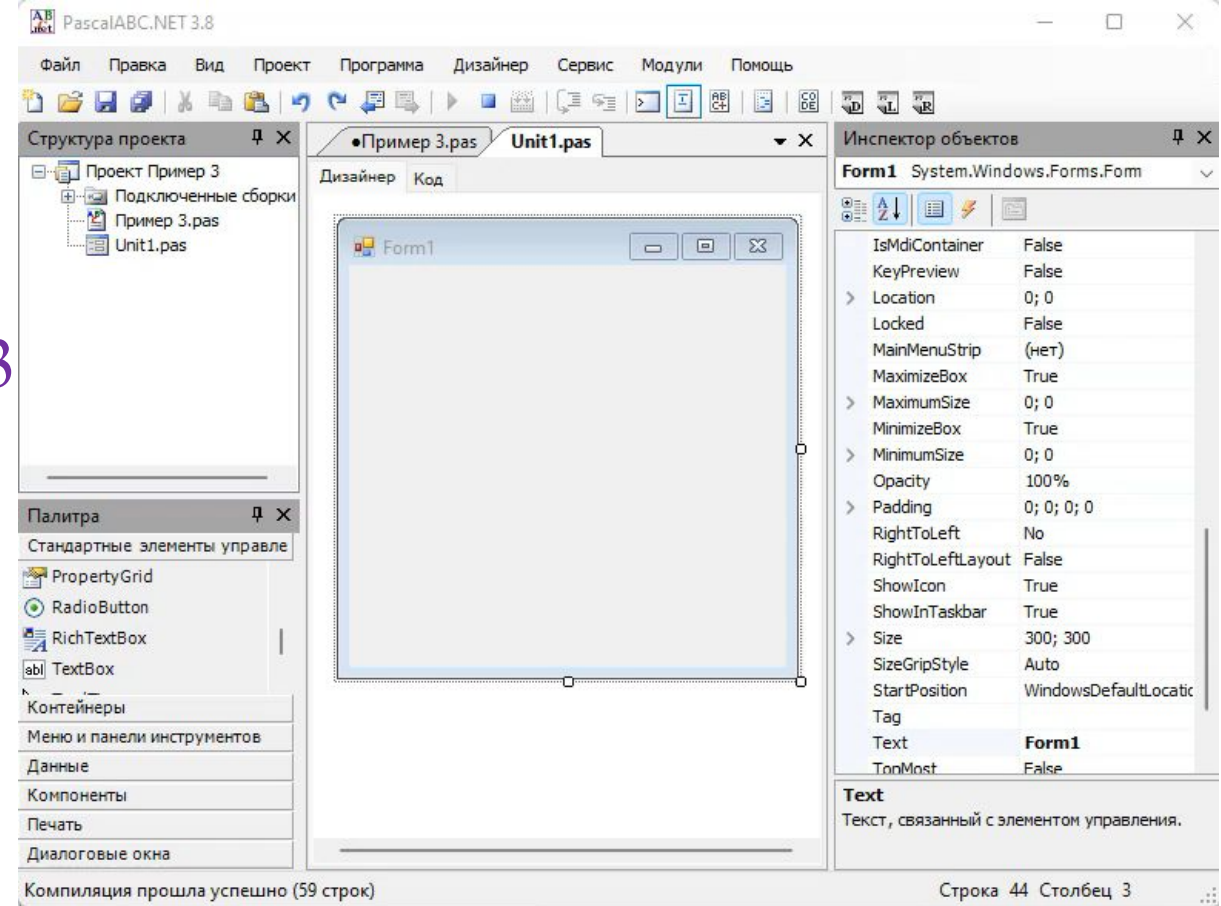
1. Изменить свойство *Text* у формы на «**Знакомство**».
2. Разместить на форме две метки и текстовое поле.
3. Изменить свойство *Text* у *label1* на «**Введите свое имя и нажмите клавишу Enter**».
4. Очистить поле свойства *Text* у *Label2*.

## Пример 3

Создать проект, в котором пользователя попросят ввести его имя, а затем, после нажатия клавиши *Enter*, будет выдано сообщение «**Имя, приятно с Вами познакомиться!**»

5. Написать обработчик события *KeyPress* для компонента *Edit1*, который будет проверять нажатие клавиши ввода (код клавиши **Enter** — 13). Если клавиша нажата, то поменять свойство *Text* у *label2*:

```
if e.KeyChar = #13 then  
label2.Text := TextBox1.T  
ext + ', приятно с Вами  
познакомиться!';
```



Текстовое поле *TextBox* используется также для ввода и вывода чисел. При этом необходимо использовать функции для преобразования строк в числа и чисел в строки. Эти функции приведены в таблице:

Название функции	Действие
Ввод с помощью Edit	
StrToInt	Преобразование строки в целое число
StrToFloat	Преобразование строки в значение с плавающей запятой
Вывод с помощью Edit	
IntToStr	Преобразование целого числа в строку
FloatToStr	Преобразование вещественного числа в строку

## Пример 4

Создать проект, в котором пользователь сможет ввести число, получить его значение в квадрате и квадратный корень из этого числа.

### *Этапы выполнения задания*

1. Изменить свойство *Text* у формы на «**Квадрат и корень**».
2. Разместить на форме три метки, три текстовых поля и кнопку.
3. Изменить свойство *Text* у *Label1* на «**Введите неотрицательное число**».
4. Изменить свойство *Text* у *Label2* на «**Квадрат**».
5. Изменить свойство *Text* у *Label3* на «**Корень**».

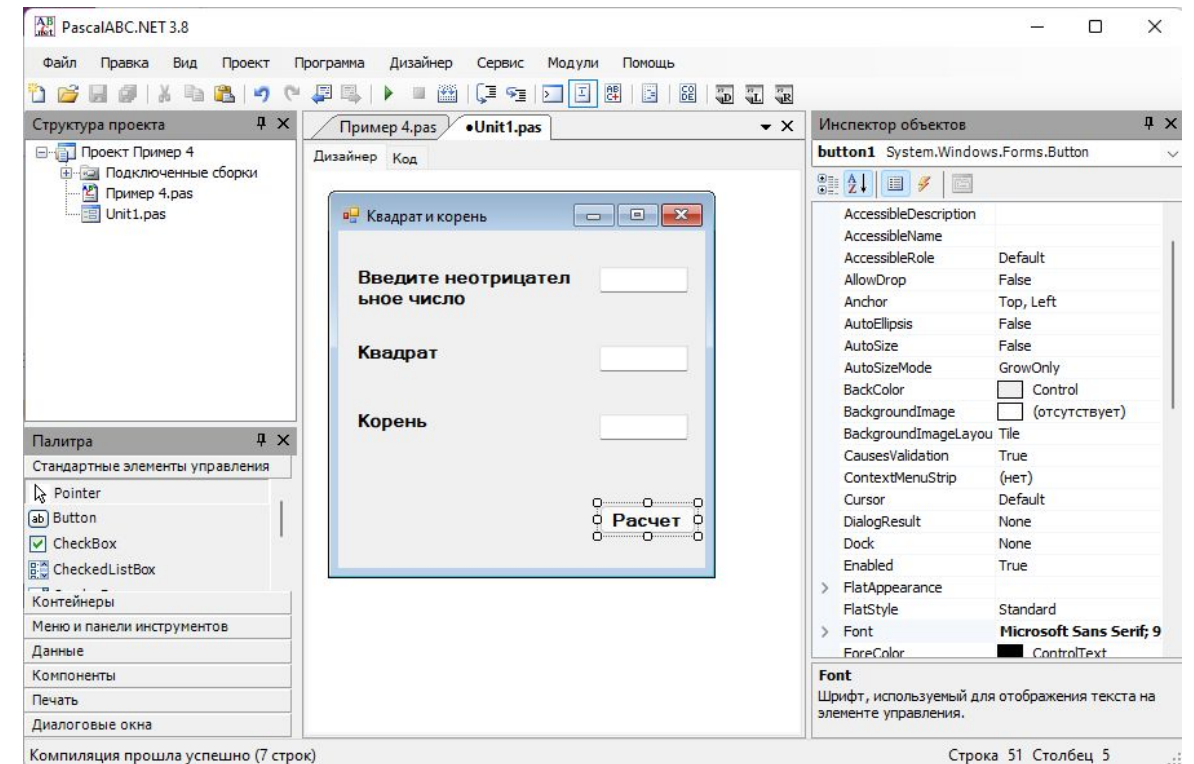
## Пример 4

Создать проект, в котором пользователь сможет ввести число, получить его значение в квадрате и квадратный корень из этого числа.

### Этапы выполнения задания

6. Изменить свойство *Text* у *Button1* на «**Расчет**».

7. Написать обработчик *Click* для кнопки.



Элементы управления флажок  
(*CheckBox*) и переключатель  
(*RadioButton*)

**Флажок** используется в приложениях для включения или выключения каких-либо опций.

На панели компонентов  
*Стандартные* элементы  
управления флажок


изображен в виде  CheckBox

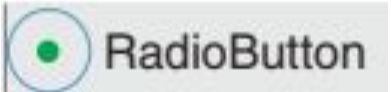
объекта — *CheckBox*.

Флажок, помещенный на

форму, получает имя

*checkBoxN*, где N — номер 1, 2, 3...

Свойство	Назначение
Checked	Значение true у этого свойства показывает, что установлена галочка —  , пустое окно индикатора —  соответствует значению false
Text	Надпись возле компонента checkBox
RightToLeft	Определяет, с какой стороны кнопки размещается надпись: Yes (надпись слева), No (надпись справа)
CheckState	Определяет состояние компонента: Unchecked — не выделен, Checked — выделен, Indeterminate (  ) — промежуточное состояние. Первые два состояния соответствуют свойству Checked

Аналогичным образом используется компонент **Переключатель** (*радиокнопка*). На панели компонентов *Standard* радиокнопка изображена в виде , имя объекта — ***RadioButton***.

Переключатель, помещенный на форму, получает имя ***radioButtonN***, где N — номер 1, 2, 3... .

Обычно радиокнопки образуют группы взаимосвязанных индикаторов, позволяющих выбрать только одну из нескольких взаимоисключающих альтернатив. При размещении на форме нескольких переключателей, включенным должен быть только один из них (контейнер ***GroupBox***).



## Пример 5

Создать проект для вычисления частного от деления одного целого числа на другое. Числа задаются в текстовых полях. Результат вычисляется при нажатии на кнопку «Частное» и помещается в третье текстовое поле. Результат зависит от состояния флажка.

### *Этапы выполнения задания*

1. Поместить на форму текстовые поля (3), надписи (3), флажок и кнопку.
2. Для компонента *textBox3* установить значение true для свойства *ReadOnly*.

## Пример 5

Создать проект для вычисления частного от деления одного целого числа на другое. Числа задаются в текстовых полях. Результат вычисляется при нажатии на кнопку «Частное» и помещается в третье текстовое поле. Результат зависит от состояния флажка.

### *Этапы выполнения задания*

3. Изменить свойство *Text* у компонентов **label** («Делимое», «Делитель», «Частное»).
4. Изменить свойство *Text* компонента **button1** на «Разделить».
5. Изменить свойство *Text* компонента **checkBox1** на «Целочисленное деление».

## Пример 5

Создать проект для вычисления частного от деления одного целого числа на другое. Числа задаются в текстовых полях. Результат вычисляется при нажатии на кнопку «Частное» и помещается в третье текстовое поле. Результат зависит от состояния флажка.

### *Этапы выполнения задания*

6. Написать обработчик события *Click* для компонента **button1.4**.

6.1. Проверить, что поля компонентов **textBox1** и **textBox2** не пусты. Иначе вывести сообщение «**Одно из полей не заполнено**».

## Пример 5

Создать проект для вычисления частного от деления одного целого числа на другое. Числа задаются в текстовых полях. Результат вычисляется при нажатии на кнопку «Частное» и помещается в третье текстовое поле. Результат зависит от состояния флажка.

### *Этапы выполнения задания*

6.2. Проверить состояние переключателя **checkBox**. Если он включен, то выполнить целочисленное деление, иначе обычное деление.

6.3. Вывести результат.

## Пример 5

Создать проект для вычисления частного от деления одного целого числа на другое. Числа задаются в текстовых полях. Результат вычисляется при нажатии на кнопку «Частное» и помещается в третье текстовое поле. Результат зависит от состояния флажка.

### *Этапы выполнения задания*

7. Выполнить программу для различных значений. Проверить работу приложения, когда одно из полей **textBox1** или **textBox2** (или оба поля) пустые.

# Повторим

Какие компоненты относят к элементам управления?

Наиболее распространенными элементами управления являются: кнопки, редактируемые поля, списки выбора, флажки, переключатели и т. д.


# Повторим

Как поместить компонент на форму?

Компонент может быть размещен путем перетаскивания его с панели Палитра на форму.

# Повторим

Какие свойства компонента *button* вы можете назвать?

Некоторые свойства кнопки Button	
Свойство	Назначение
Text	Заголовок (внешнее имя) кнопки, текст, который отображается на кнопке. По умолчанию — button1
Font	Свойства шрифта для подписи заголовка. Свойство Font является сложным, о чем свидетельствует значок  , при нажатии на который раскрываются все свойства шрифта
Height	Высота кнопки
Weight	Ширина кнопки

Left	Горизонтальная координата положения верхнего левого угла кнопки на форме
Top	Вертикальная координата положения верхнего левого угла кнопки на форме
(Name)	Имя (внутреннее) кнопки. Используется в программном коде для обращения к объекту. Является идентификатором
Enabled	Значение True этого свойства обеспечивает доступность кнопки для мыши или клавиатуры
Visible	Значение True этого свойства обеспечивает видимость кнопки во время выполнения приложения



# Повторим

Какое событие является основным для компонента *button*?

Основным событием кнопки является *Click*.

# Повторим

Для чего предназначен компонент *label*?

Компонент *label* (*метка*) предназначен для отображения текста на форме.

# Повторим

В каких случаях используется компонент TextBox?

**Текстовое поле** — компонент, который предназначен для ввода и вывода текстовой информации.

Текстовое поле *TextBox* используется также для ввода и вывода чисел.

# Повторим

Для чего предназначены компоненты *checkBox* и *radioButton*?

Элемент управления *checkBox* (флажок) и *radioButton* (переключатель) используются в приложениях для включения или выключения каких-либо опций.

# Домашнее задание

§ 3