

**Лабораторная работа №15 по дисциплине
"Вычислительная техника и программирование"**

ТЕМА: ПЕРЕМЕЩЕНИЕ ОКОН И КОМПОНЕНТОВ.

СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ПАПЕК

1. В Вашей папке создайте папку с именем ВТП_лр15, совпадающим с именем файла с этим заданием, и затем скопируйте файл с заданием в эту папку.

2. При выполнении задания создавайте в папке ВТП_лр15 для каждой программы папку с именем, по смыслу соответствующим действиям, выполняемым составляемой программой, и сохраняйте все файлы, имеющие отношение к этой программе, в этой созданной папке (имена сохраняемых файлов должны отличаться от имен, присваиваемых по умолчанию, и должны по смыслу соответствовать действиям, выполняемым составляемой программой, причем рекомендуется, чтобы имя модуля отличалось от имени проекта цифрой, соответствующей номеру версии модуля).

3. При создании приложений изменяйте имена в заголовках окон этих приложений и имена объектов в этих окнах так, чтобы имена **отличались от стандартных имен** (Form1, Button1 и т.д.) и соответствовали назначению этих окон и объектов.

4. ПРИ ВЫПОЛНЕНИИ СЛЕДУЮЩЕЙ ЛАБОРАТОРНОЙ РАБОТЫ ДЕЙСТВУЙТЕ АНАЛОГИЧНО ПРЕДЫДУЩЕМУ.

ПЕРЕМЕЩЕНИЕ ОКОН И КОМПОНЕНТОВ

5. Форма (компонент Form) является основой для размещения различных компонентов (кнопок, текстовых полей, надписей, списков, меню и прочих), с использованием которых формируется пользовательский интерфейс приложения.

Ниже перечислены основные свойства формы и приведены краткие описания этих свойств.

Свойство	Описание
Name	Имя формы (обычно Form1 или Form2 , Form3 и т.д.). В операторах программы используется для доступа к свойствам и методам самой формы или свойствам и методам компонентов этой формы. Для одной из форм приложения, которая является главной, имя формы в операторах можно не указывать (если форма в приложении одна, то она автоматически является главной). Например, для главной формы, запись Form1.Color равносильна записи Color , запись Button1.Caption равносильна записи Form1.Button1.Caption . В простых приложениях (с одной формой) значение свойства Name обычно не меняют, и оно остается равным Form1 .
Caption	Текст в заголовке окна формы, который по умолчанию совпадает со значением свойства Name – например, Form1 , но затем обычно заменяется на другое значение, соответствующее назначению приложения.
Icon	Задаёт значок окна формы в его левом верхнем углу при несвернутом окне приложения.
Color	Определяет цвет клиентской области формы. По умолчанию значение свойства равно clBtnFace .
Enabled	Признак доступности формы (в том числе для выполнения действий с

	размещенными на ней компонентами) во время выполнения приложения (если значение свойства равно True , то форма доступна, если False – то недоступна).
Align	<p>Определяет выравнивание формы внутри экрана.</p> <p>По умолчанию значение свойства равно alNone, и при запуске приложения окно формы без изменения размеров остается на экране там же, где оно было размещено во время проектирования. При этом значении свойства нет ограничений на изменение положения формы на экране и на изменение ее размеров.</p> <p>При значениях свойства alTop или alBottom форма занимает по ширине всю верхнюю или, соответственно, всю нижнюю часть Рабочего стола без возможности изменения ширины формы.</p> <p>При значениях свойства alLeft и alRight форма занимает по высоте всю левую или, соответственно, всю правую часть рабочего стола без возможности изменения высоты формы.</p>
Top	Расстояние в пикселях от верхнего края экрана до верхнего края формы.
Left	Расстояние в пикселях от левого края экрана до левого края формы.
Position	Определяет размер и положение окна. Если, например, значение свойства равно poScreenCenter или poDesktopCenter , то окно будет находиться в центре экрана.
Height	Высота формы в пикселях.
Width	Ширина формы в пикселях.
ClientHeight	Высота клиентской области формы в пикселях (без учета высоты заголовка и толщины нижней границы формы). Компоненты могут быть размещены только в клиентской области формы.
ClientWidth	Ширина клиентской области формы в пикселях (без учета толщины боковых границы формы).
WindowState	Определяет вид окна формы на экране. Принимает значения, соответствующие нормальному виду окна (wsNormal), свернутому окну (wsMinimized), развернутому окну (wsMaximized) приложения.
Visible	Признак видимости формы на экране во время выполнения приложения. Если значение свойства равно True , то форма является видимой, если False – невидимой.
Hint	Текст (подсказка), который появляется рядом с указателем "мыши", указывающим на форму во время выполнения приложения (для появления текста необходимо, чтобы значение свойства ShowHint было равно True).
ShowHint	Признак разрешения появления текста (подсказки), определяемого значением свойства Hint . Если значение свойства ShowHint равно True , то текст появляется, если False – не появляется.

6. При создании приложений, при выполнении которых изменяется вид окна приложения, его размеры и положение на экране, в процедурах обработки событий должны находиться операторы, изменяющие значения таких, например, свойств формы, как **Color**, **Height**, **Width**, **Top**, **Left**, **Position** и других.

Примеры записи операторов, изменяющих значения свойств формы:

```
Form1.Color:=clRed;      //Изменение цвета формы на красный
Form1.Height:=Form1.Height div 2;  //Уменьшение высоты формы в 2 раза
Form1.Top:=10;          //Перед названием свойства указано имя формы
Left:=20;               //Перед названием свойства можно не указывать имя главной
                        // формы
```

Свойство **Color**, определяющее цвет, может принимать значения из множества специальных констант типа **TColor** (значения этих констант можно выбирать в окне **Object Inspector** из раскрывающегося списка в строке с названием свойства **Color**). Такие константы использовались, например, в операторе процедуры обработки события в созданном выше приложении (см. Рисунок 18.4). Значения свойства **Color** можно задавать также в виде шестнадцатеричных чисел (например, значению константы **clRed**, определяющей чистый красный цвет, соответствует число **\$000000FF**, чистому зеленому цвету соответствует число **\$0000FF00**, желтому цвету – число **\$0000FFFF**, чистому синему цвету – число **\$00FF0000**, белому цвету – число **\$00FFFFFF**, черному цвету – число **\$00000000**).

Значения свойств **Top** и **Left** определяют, соответственно, расстояние в пикселях от верхнего края экрана до верхнего края формы и от левого края экрана до левого края формы, или, иначе говоря, определяют положение на экране левого верхнего угла формы. При изменении положения на экране окна приложения можно рассматривать это окно как прямоугольник, изменяющий свое положение внутри прямоугольника большего размера (экрана). При этом высота и ширина экрана (рассматриваемого как компонент с именем **Screen**) определяются, соответственно, значениями его свойств **Screen.Height** и **Screen.Width**. Например, для того, чтобы во время выполнения приложения его окно оказалось вплотную к краям экрана в левом верхнем углу экрана, в одной из процедур обработки событий могут, например, присутствовать операторы

```
Top:=0; Left:=0;
```

чтобы это окно оказалось в правом верхнем углу экрана – операторы

```
Top:=0; Left:=Screen.Width-Width; //Width - ширина окна приложения
```

чтобы это окно оказалось в правом нижнем углу экрана – операторы

```
Top:=Screen.Height-Height; Left:=Screen.Width-Width;  
//Height - высота, Width - ширина окна приложения
```

чтобы это окно оказалось в левом нижнем углу экрана – операторы

```
Top:=Screen.Height-Height; Left:=0;
```

Для того чтобы во время выполнения приложения его окно оказалось в центре экрана, в одной из процедур обработки событий может, например, присутствовать оператор `Position:=poScreenCenter;`

7. Наиболее характерным событием для формы является событие **OnCreate**, наступающее при создании формы. Процедура обработки этого события будет создана, если поместить указатель "мыши" в область формы или на ее значок в окне **Structure** и затем выполнить двойной щелчок левой кнопкой "мыши". Эта процедура обычно используется для установки начальных значений и выполнения других подготовительных действий при запуске приложения.

Для формы можно также использовать событие **OnClick** (соответствует щелчку левой кнопкой "мыши", когда ее указатель находится в клиентской области формы), событие **OnDblClick** (соответствует двойному щелчку левой кнопкой "мыши", когда ее указатель находится в клиентской области формы) и другие события.

8. Все без исключения компоненты (включая форму) имеют целочисленное свойство **Tag**, первоначальное значение которого по умолчанию равно нулю. Это свойство представляет собой по сути дела целочисленную переменную, связанную с компонентом. Значение этого свойства может быть использовано, например, как номер состояния компонента (этому состоянию может соответствовать, например, определенный набор значений свойств этого компонента), в зависимости от значения которого (номера) в программе могут быть выполнены те или иные действия.

ПРОГРАММА 15-1

Создайте приложение, при запуске которого в центре экрана появляется окно формы небольшого размера стандартного цвета.

Затем, при каждом выполнении щелчка левой кнопкой "мыши" в области формы, эта форма должна сначала переместиться в левый верхний угол экрана и одновременно изменить свой цвет, затем при выполнении щелчков форма должна перемещаться по оставшимся углам экрана и одновременно менять свой цвет, а также свои высоту и ширину (сначала в 2, затем в 3, затем в 4 раза) и затем снова должна "вернуться" в левый верхний угол с теми цветом и размерами, которые были у нее в этом углу.

При нажатии на любую клавишу на клавиатуре (событие **OnKeyPress**) форма должна она оказаться с первоначальными цветом и размерами в центре экрана (значение свойства **Position** формы должно быть равно **poScreenCenter**).


Форма должна перемещаться по углам экрана **против часовой стрелки** при создании приложения на компьютере с **нечетным номером** и **по часовой стрелке** при создании приложения на компьютере с **четным номером**.

Окно формы должно быть полностью видно на экране при всех перемещениях.

Используйте для определения положения окна формы на экране ее свойства **Top** и **Left**, для определения размеров окна формы – ее свойства **Height** и **Width**, для хранения номера состояния формы – ее свойство **Tag**.

При использовании свойств формы **Form1** наименования этих свойств можно указывать как вместе с именем формы **Form1**, так и без него, например: **Form1.Height** или просто **Height**.

Размеры экрана (объект **Screen**) определяются его свойствами **Screen.Height** и **Screen.Width**.

9. Кнопка **TButton**  с вкладки **Standard** окна **Tool Palette** является простейшей и наиболее употребительной из кнопок.

Ниже перечислены основные свойства кнопки **TButton** и приведены краткие описания этих свойств.

Свойство	Описание
Name	Имя кнопки (обычно Button1 или Button2 , Button3 и т.д.). В операторах программы используется для доступа к свойствам и методам кнопки, например: <code>Button1.Height:=Button1.Height *2;</code> <code>Button1.Width:=Button1.Width div 2;</code> В простых приложениях значение свойства Name обычно не меняют, и оно остается равным Button1 или Button2 , Button3 и т.д..
Caption	Текст на кнопке (или, иначе говоря, ее заголовок), который по умолчанию совпадает со значением свойства Name – например, Button1 , но затем обычно заменяется на другое значение, соответствующее назначению кнопки, например: <code>if Button1.Caption='Пуск' then Button1.Caption='Стоп'</code> <code>else Button1.Caption='Пуск';</code> Для кнопки можно указать символ быстрого доступа. Для этого значение свойства Caption должно содержать перед этим символом символ амперсанда (&). Следующий за амперсандом символ будет выгледеть подчеркнутым и будет являться символом быстрого доступа (в этом случае доступ к команде может быть получен при одновременном нажатии клавиши Alt и клавиши этого символа). Для использования символа амперсанда в значении свойства Caption этот символ следует ввести дважды подряд(&&).
Font	Шрифт текста на кнопке (за исключением цвета шрифта).
Enabled	Признак доступности кнопки. Если значение свойства равно True , то кнопка

	доступна, если False – то недоступна (например, в результате щелчка по кнопке событие OnClick не наступает).
Top	Расстояние в пикселях от верхнего края клиентской части формы до верхнего края кнопки.
Left	Расстояние в пикселях от левого края клиентской части формы до левого края кнопки.
Height	Высота кнопки в пикселях.
Width	Ширина кнопки в пикселях.
Visible	Признак видимости кнопки. Если значение свойства равно True , то кнопка является видимой, если False – невидимой.
Hint	Текст (подсказка), который появляется рядом с указателем "мыши", указывающим на кнопку (для появления текста необходимо, чтобы значение свойства ShowHint было равно True).
ShowHint	Признак разрешения появления текста (подсказки), определяемого значением свойства Hint . Если значение свойства ShowHint равно True , то текст появляется, если False – не появляется.

10. Наиболее характерным свойством кнопки **TButton** является свойство **Caption**, наиболее характерным событием – **OnClick**, наступающее при щелчке левой кнопкой "мыши", когда ее указатель находится в области кнопки (процедура обработки этого события будет создана, если поместить указатель "мыши" в область кнопки или на ее значок в окне **Structure** и затем выполнить двойной щелчок левой кнопкой "мыши").

11. Значения свойств **Top** и **Left** определяют, соответственно, расстояние в пикселях от верхнего края клиентской области формы до верхнего края кнопки и от левого края клиентской области формы до левого края кнопки, или, иначе говоря, определяют положение в клиентской области формы левого верхнего угла кнопки. При изменении положения кнопки ее можно рассматривать как прямоугольник, изменяющий свое положение внутри прямоугольника большего размера (клиентской области формы). При этом высота и ширина клиентской области формы определяются, соответственно, значениями ее свойств **ClientHeight** и **ClientWidth**. Например, для того, чтобы во время выполнения приложения кнопка оказалась расположенной вплотную к краям клиентской области формы в левом верхнем углу этой области, в одной из процедур обработки событий могут, например, присутствовать операторы

```
Button1.Top:=0; Button1.Left:=0;
```

чтобы кнопка оказалась в правом нижнем углу клиентской области формы – операторы

```
Button1.Top:=ClientHeight-Button1.Height;
```

```
Left:=ClientWidth-Button1.Width;
```

чтобы кнопка оказалась в центре клиентской области формы – операторы

```
Button1.Top:=ClientHeight div 2-Button1.Height div 2;
```

```
Left:=ClientWidth div 2-Button1.Width div 2;
```

или операторы

```
Button1.Top:=(ClientHeight-Button1.Height) div 2;
```

```
Left:=(ClientWidth-Button1.Width) div 2;
```

ПРОГРАММА 15-2

Создайте приложение, при запуске которого в центре клиентской области формы стандартного цвета находится кнопка **TButton** небольшого размера.

Затем, при нажатии на кнопку, эта кнопка должна сначала переместиться вплотную в левый верхний угол клиентской области формы (то есть верхний и левый края формы должны вплотную примыкать к соответствующим краям клиентской области формы).

Одновременно с перемещением кнопки форма должна изменить свой цвет.

При следующих нажатиях на кнопку эта кнопка должна перемещаться вплотную в оставшиеся углы клиентской области формы и одновременно увеличивать свои высоту и ширину (сначала в 2, затем в 3, затем в 4 раза) и затем снова должна "вернуться" в левый верхний угол клиентской области формы с теми размерами, которые были у нее в этом углу.

Одновременно с перемещениями кнопки форма должна менять свой цвет.

Все эти перемещения должны правильно выполняться независимо от размеров формы и кнопки, то есть для определения положения и размеров кнопки следует использовать выражения с названиями свойств, а не численные значения этих свойств (за исключением определения положения кнопки в левом верхнем углу клиентской области формы).

При щелчке левой кнопкой "мыши" в области формы вне кнопки кнопка должна оказаться в центре клиентской области формы с первоначальными размерами, а форма должна приобрести первоначальный цвет.

Кнопка должна перемещаться по углам клиентской области формы **против часовой стрелки** при создании приложения на компьютере с **нечетным номером** и **по часовой стрелке** при создании приложения на компьютере с **четным номером**.

Используйте для определения положения кнопки в форме ее свойства **Top** и **Left**, для определения размеров кнопки – ее свойства **Height** и **Width**, для хранения номера состояния кнопки – свойство **Tag** кнопки или формы.

Размеры клиентской области формы, доступной для перемещения кнопки, определяются значениями свойств формы **ClientHeight** и **ClientWidth**.

ОТЧЕТ О ВЫПОЛНЕНИИ РАБОТЫ

12. Продемонстрируйте проекты созданных приложений, а также их выполнение в среде программирования и независимо от нее.
