

Курс лабораторных работ по прикладному программированию

1. Л.Р. 1 (2 часа): Основы программирования в среде C++Builder: запуск Builder; назначение компонентов главного окна: панель горячих кнопок, палитры компонентов, галерея шаблонов форм, окно инспектора объектов, окно формы, окно модуля. Проектирование оболочки программы-приложения: основное меню (**MainMenu**), всплывающее меню (**PopUpMenu**), их свойства, текстовые надписи на полотне.

На полотне формы (Form1) разместить компонент **MainMenu** палитры компонентов **Standard** (двойной щелчок мыши на пиктограмме компонента). Открыть проектировщик меню (двойной щелчок мыши на компоненте **MainMenu1**). Заполнить пункты меню **Caption** (Заголовок) окне инспектора объектов (Object Inspector) :

&Ввод данных и дочерние меню:

&Ввод
&Контроль
&Диалог
&Расчет
&Выполнить
&Просмотр
&График
&Построить
в&Ыход

Листинг. 1.

(&Ы - означает инициализацию данного пункта меню дополнительно с помощью комбинации клавиш Alt+ы).

- Обработка события для меню **в&Ыход** (двойной щелчок на компоненте меню):

```
{  
  Close();  
}
```

Листинг. 2.

- Обработка события для остальных меню:

```
{  
  ShowMessage("Простое сообщение");  
}
```

Листинг. 3.

- Поместить на полотне формы компонент **Static Text** палитры компонентов **Additional** (в ранних версиях C++Builder данного компонента нет). Изменить свойство **StaticText1** "Caption" на *Нажмите правую клавишу мыши*.

- Компонент **PopupMenu1** :

&Нарисовать

&Очистить

&Изменить цвет фона

Изменить &цвет линии

Изменить &тип линии

Изменить тол&щину линии

Листинг . 4 .

- Подключить всплывающее меню **PopupMenu1** к полотну объекта Form1. Для этого щелкнуть левой клавишей мыши на полотне Form1 (в окне инспектора объектов заглавная надпись изменится на *Form1:TForm*); найти свойство *PopupMenu* и изменить его значение на *PopupMenu1*.
- Сохранить проект и файлы , протранслировать проект.

Л.Р. 2 (2 часа): графика на полотне Формы (объект Canvas, его свойства) и очистка полотна (кнопки "Построить" и "Очистить"), изменение цветовой палитры элементов графики (компонент **ColorDialog**).

- Обработка события для меню **&Нарисовать** :

```
{
  Canvas->MoveTo(50,50);
  Canvas->LineTo(ClientWidth-50,50);
  Canvas->LineTo(ClientWidth-50,ClientHeight-50);
  Canvas->LineTo(50,ClientHeight-50);
  Canvas->LineTo(50,50);
}
```

Листинг . 5 .

или (результат будет почти такой же*):

```
{
  Canvas->Rectangle( 100,100, Form1->ClientWidth-50, Form1->ClientHeight-50);
}
```

Листинг . 6 .

- Обработка события для меню **&Очистить**:

```
{
  Refresh();
}
```

Листинг . 7 .

- **Изменение цветовой палитры** : Поместить на полотно формы Form1 компонент **ColorDialog** из палитры компонентов **Dialogs**.
- Обработка события для меню **&Изменить цвет фона**:

```
{
```

```

ColorDialog1->Color = Form1->Color;
if (ColorDialog1->Execute())
    Form1->Color = ColorDialog1->Color;
}

```

Листинг. 8.

* Метод, приведенный в листинге 6, в отличие от предыдущего, заполняет область внутри построенного прямоугольника в стиле *Canvas->brush->Style* цветом *Canvas->brush->Color*.

- Обработка события для меню **Изменить &Цвет линии**:

```

{
ColorDialog1->Color = Form1->Color;
if (ColorDialog1->Execute())
    Form1->Color = ColorDialog1->Color;
}

```

Листинг. 9.

Л.Р. 3 (2 часа): графика на полотне Формы, контейнер для компонент (компонент **Panel**), визуализация окон и панелей, кнопки (**Button**, **BitBtn**), комбинированное диалоговое окно (**ComboBox**), обработка событий для кнопок.

- Создание диалога для пункта меню " Изменить &Тип линии ":
- Поместить на полотно формы Form1 компонент **Panel** из палитры компонент **Standard**.
- Изменить свойство **Panel1** "Visible" на *False*.
- Обработка события для меню **Изменить &Тип линии**.

```

{
Panel1->Visible=true;
}

```

Листинг. 10.

- Поместить на Панель1 компонент **ComboBox** (комбинированное диалоговое окно) из палитры компонент **Standard** и два компонента **BitBtn** (стандартная кнопка с растровым значком) из палитры компонент **Additional**.
- В окне инспектора объектов для **ComboBox1** изменить свойство "Style" на *csDropDownList* (ввод текста только из списка).
- В окне инспектора объектов для **ComboBox1** найти свойство "Items" и нажать на кнопку Набрать в окне редактора списка строк (String List Editor) следующие строки:

```

psSolid
psDash
ps Dot
psDashDot

```

psDashDotDot
psClear
psInsideFrame

Листинг. 11.

(этот список будет отображаться в "падающей" части ListBox компонента ComboBox)

- Изменить свойство "Kind" для компонента **BitBtn1** на *bkOK*
- Изменить свойство "Kind" для компонента **BitBtn2** на *bkCancel*
- Обработка события для кнопки **ОК** (сохранить изменения)

```
{
switch (ComboBox1->ItemIndex)
{
case 0: { Canvas->Pen->Style = psSolid;          break; }
case 1: { Canvas->Pen->Style = psDash;          break; }
case 2: { Canvas->Pen->Style = psDot;           break; }
case 3: { Canvas->Pen->Style = psDashDot;       break; }
case 4: { Canvas->Pen->Style = psDashDotDot;    break; }
case 5: { Canvas->Pen->Style = psClear;         break; }
case 6: { Canvas->Pen->Style = psInsideFrame;   break; }
}
Panel1->Visible=false;
}
```

Листинг. 12.

- Обработка события для кнопки **Cancel** (отменить изменения)

```
{
Canvas->Pen->Width = 1;
Panel2->Visible = false;
Refresh();
}
```

Листинг. 13.

Л.Р. 4 (2 часа): создание диалога, поля редактирования (**Edit**), создание собственных процедур, подключение других модулей, создание дочерних меню для всплывающего меню.

- Создание диалога для пункта меню "Изменить тол&Щину линии":
- Поместить на полотно формы Form1 компонент **Panel** из палитры компонентов **Standard**.

- Изменить свойство **Panel2** "Visible" на *False*.
- Поместить на Панель2 компонент **Edit** (поле редактирования) из палитры компонентов **Standard** и два компонента **BitBtn** из палитры компонентов **Additional**.
- Изменить свойство "Kind" для компонента **BitBtn1** на *bkOK*.
- Изменить свойство "Kind" для компонента **BitBtn2** на *bkCancel*.
- Обработка события для меню **Изменить тол&щину линии**:

```
{
Panel2->Visible = true;
}
```

Листинг. 14.

- Обработка события для кнопки **OK** (сохранить изменения):

```
{
Canvas->Pen->Width = StrToInt(Edit1->Text);
Panel2->Visible = false;
draw_pkm_97(0);
}
```

Листинг. 15.

- Обработка события для кнопки **Cancel** (отменить изменения):

```
{
Canvas->Pen->Width = 1;
Panel2->Visible = false;
Refresh();
draw_pkm_97(0);
}
```

Листинг. 16.

- Оформить блок рисования (см. Листинг. 5, 6) в виде процедуры:

```
void __fastcall TForm1::draw_pkm_97(TObject *Sender)
{
Canvas->Rectangle( 100,100, Form1->ClientWidth-50, Form1->ClientHeight-50);
}
```

Листинг. 17.

- Поместить ее в модуль **unit1** после операторов

```
TForm1 *Form1;
```

Листинг. 18.

- В блоке описания процедур (**unit1.h**) добавить:

```
void __fastcall draw_pkm_97(TObject *Sender);
```

Листинг. 19.

- Добавить вызов процедуры в обработчики событий изменения цвета и типа линии (Листинг. 9, 12):

```
draw_pkm_97(0);
```

- Добавить в компонент **PopupMenu1** :

&Заполнение...

и создать дочерние меню:

ти&n

цв&ет

Для этого нажать правую клавишу мыши и выбрать из предложенного меню пункт:

Create Submenu (создать подменю).

- Обработка события для подменю **цв&Ет**:

```
{
ColorDialog1->Color = Form1->Canvas->Brush->Color;
if (ColorDialog1->Execute())
{
Form1->Canvas->Brush->Color = ColorDialog1->Color;
draw_pkm_97(0);
}
}
```

Л.Р. 5 (2 часа): графика на полотне Формы, заполнение замкнутых областей (класс Brush, его свойства), контейнер компонентов **GroupBox**, компонент **UpDown** (кнопка со стрелками для изменения числовой величины).

- Создание диалога для пункта подменю " ти&П ":
- Поместить на полотно формы Form1 компонент **GroupBox** из палитры компонентов **Standard**
- Изменить свойство **GroupBox1** "Visible" на *False*
- Изменить свойство **GroupBox1** "Caption" на *Выберите тип заполнения*
- Поместить на GroupBox1 компонент **ComboBox** (комбинированное диалоговое окно) из палитры компонентов **Standard** и две компоненты **BitBtn** (из палитры компонентов **Additional**
- В окне инспектора объектов для **ComboBox2** изменить свойство "Style" на *csDropDownList* (ввод текста только из списка)
- В окне инспектора объектов для **ComboBox2** найти свойство "Items" и нажать на кнопку **...** . Набрать в окне редактора списка строк (String List Editor) следующие строки:

Сплошная заливка

Очистка

Штриховка +45

Штриховка -45

Штриховка 0+90

Штриховка +45-45

Штриховка 0 (горизонтальная)

Штриховка 90 (вертикальная)

Листинг. 24.

- Изменить свойство "Kind" для компоненты **BitBtn1** на *bkOK*
- Изменить свойство "Kind" для компоненты **BitBtn2** на *bkCancel*
- Обработка события для подменю **ти&п**

```
{  
GroupBox1->Visible = true;  
}
```

Листинг. 25.

- Обработка события для кнопки **ОК** (сохранить изменения) компоненты **GroupBox1**

```
{  
switch (ComboBox2->ItemIndex)  
{  
case 0: Canvas->Brush->Style = bsSolid; break;  
case 1: Canvas->Brush->Style = bsClear; break;  
case 2: Canvas->Brush->Style = bsBDiagonal; break;  
case 3: Canvas->Brush->Style = bsFDiagonal; break;  
case 4: Canvas->Brush->Style = bsCross; break;  
case 5: Canvas->Brush->Style = bsDiagCross; break;  
case 6: Canvas->Brush->Style = bsHorizontal; break;  
case 7: Canvas->Brush->Style = bsVertical; break;  
}  
GroupBox1->Visible = false;  
draw_pkm_97(0);  
}
```

Листинг. 26.

- Обработка события для кнопки **Cancel** (отменить изменения) компоненты **GroupBox1**

```
{  
GroupBox1->Visible = false;  
}
```

Листинг. 27.

- Использование компонента **UpDown** (кнопка со стрелками для изменения числовой величины).:
- На полотне панели **Panel2** разместить компонент **UpDown** из палитры компонентов **Win95**
- Изменить свойство "Associate" для компонента **UpDown1** на *Edit1*
- Изменить свойство "Min" для компоненты **UpDown1** на *1*.
- Изменить свойство "Max" для компоненты **UpDown1** на *100*.
- Изменить свойство "Increment" для компоненты **UpDown1** на *1*.

Л.Р. 6 (2 часа): новая Форма, окно диалога, метки (Label), поля редактирования (Edit)

- Открыть новую форму (Form2), используя меню **File/New**. Выбрать из многостраничной палитры шаблонов форм галерею **Dialog**, а из этой галерея шаблон **Standard Dialog**. Полученное окно диалога получит имя *OKBottomDlg1*.
- Разместить на полотне *Bevell* вертикально пять меток **Label** из палитры компонентов **Standard** и подписать их соответственно, изменяя свойство *Caption* :

Размер области по X, pixel :

Размер области по Y, pixel :

Число элементов по X :

Число элементов по Y :

Базовый размер элемента :

Листинг. 28.

- Разместить на полотне *Bevell* вертикально пять полей редактирования *Edit* из палитры компонентов **Standard** , им будут автоматически присвоены имена *Edit1, ..., Edit5*. Задать начальные значения для этих полей в свойстве *Text* соответственно 400, 400, 5, 5, 20.
- Рядом с полями *Edit* разместить пять компонент **UpDown** из палитры компонентов **Win95** и "привязать" их к полям *Edit1, ..., Edit5*.
- Обработка события для кнопки **OK** :

```
{
  xa = StrToInt(x_area->Text);
  ya = StrToInt(y_area->Text);
  nx_el = StrToInt(nx->Text);
  ny_el = StrToInt(ny->Text);
  nnn = StrToInt(size_el->Text);
  OKBottomDlg->Close();
  Form1->draw_area(xa, ya, nx_el, ny_el, nnn);
}
```

Листинг. 29.

- Обработка события для кнопки **Cancel** :

```
{
  OKBottomDlg->Close();
}
```

Листинг. 30.

- “Привязать” открытие окна диалога к одному из пунктов основного меню, например:

Ввод \ Диалог

- Обработка события для меню *Диалог*:

```
{  
OKBottomDlg->Show(); // диалог для рисования периодической структуры  
}
```

Листинг. 31.

- В модуле *Unit1* в блоке описать подключение модуля *Unut2*:
#include "Unit2.h"
- В модуле *Unit2* (соответствующем окну диалога *BtnBottomDlg*) описать подключение модуля *Unut1*: #include "Unit1.h"

Листинг. 32.

-
- В модуле *Unit1.h* описать заголовок процедуры рисования перед операторами описания других процедур:

```
void __fastcall draw_area(int xa, int ya, int nx_el, int ny_el, int nnn);
```

Листинг. 33.

- В модуле *Unit1.cpp* написать текст процедуры рисования:

```
void __fastcall TForm1::draw_area(int xa, int ya, int nx_el, int ny_el, int nnn)  
{  
int i,j,x1,y1,x2,y2,x_step,y_step;  
Form1->Refresh();  
Form1->ClientWidth = xa;  
Form1->ClientHeight = ya;  
x_step=xa / nx_el; y_step = ya / ny_el;  
for (i=1; i<=nx_el; i++)  
{  
x1=(i-1)*x_step + x_step / 2 - nnn ;  
x2=x1+nnn ;  
for (j=1; j<=ny_el; j++)  
{  
y1=(j-1)*y_step + y_step / 2 - nnn ;  
y2=y1+nnn ;  
Form1->Canvas->Ellipse(x1,y1,x2,y2); // Для рисования окружностей и  
ЭЛЛИПСОВ  
// { form1->Canvas->polygon ( [ Point(x1,y1), Point(x2,y2), Point(x3,y3),...,  
Point(xn,yn) ] ); // Для рисования замкнутых многоугольников  
}  
}  
}
```

Листинг. 34.

Л.Р. 7 (1 час): контроль данных, изменение цвета визуальных компонент.

- Добавить в окне диалога *OKBottomDlg* кнопку **Button** из палитры компонентов **Standard** и подписать ее: "Контроль".
- Добавить в окне диалога *OKBottomDlg* метку **Label** из палитры компонентов **Standard** и подписать ее: "Контроль не производился".
- Обработка события для кнопки **Контроль**:

```
{ int x_k,y_k;  
  xa = StrToInt(x_area->Text);  
  ya = StrToInt(y_area->Text);  
  nx_el = StrToInt(nx->Text);  
  ny_el = StrToInt(ny->Text);  
  nnn = StrToInt(size_el->Text);  
  x_k = nnn*2* nx_el;  
  y_k = nnn*2* ny_el;  
  if (x_k > xa || y_k > ya)  
  {  
    Label6->Caption = "Данные некорректны";  
    Label6->Color = clRed ;  
  }  
  else  
  {  
    Label6->Caption = "Данные корректны";  
    Label6->Color = clGreen;  
  } }
```

ЛИСТИНГ. 35.